# Impact of CSI Feedback Strategies on LTE Downlink and Reinforcement Learning Solutions for Optimal Allocation

Alessandro Chiumento*†, Claude Desset*, Sofie Pollin†*, Liesbet Van der Perre*†, Rudy Lauwereins*†
*Interuniversity Micro-Electronics Center (IMEC) vzw Kapeldreef-75, Leuven, B-3001, Belgium
†KU Leuven, Department of Electrical Engineering - ESAT, Leuven Belgium
Email: {chiument, desset, pollins, vdperre, lauwereins}@imec.be

***Abstract*—The constant increase in wireless handheld devices and the prospect of billion of connected machines has brought the cellular community to research many different technologies able to deliver high datarate and quality of service to the mobile users. One of the problems, usually overlooked by the community, is that more devices means higher signalling necessary to coordinate transmission and to allocate resources effectively. Particularly, channel state information of the users' channels is necessary in order for the base station to assign frequency resources. On the other hand, this feedback information comes at a cost of uplink bandwidth which is traditionally not considered. In this work, we analyse the impact that reduced user feedback information has on an LTE network. A model, which considers the trade-off between downlink performance and uplink overhead is presented. We introduce different feedback allocation strategies, which follow the same structure as the ones in the LTE standard, and study their effects on the network for varying number of users and different resource allocation strategies. We show that dynamically allocating feedback resources can be beneficial for the network. In order for the base station to determine which feedback allocation strategy is the most beneficial, in specific network conditions, we propose two reinforcement learning algorithms. The first solution allows the base station to allocate one homogeneous feedback strategy valid for all the users served, while, the second more complex solution determines a different strategy for each user based on its channel conditions. The reinforcement learning methods show that, even in dynamic scenarios, each base station is capable of determining an optimal operating point autonomously, hence optimally balancing feedback overhead and benefits.**

## I. Introduction

THE proliferation of radio-capable hand-held devices in recent years has impacted wireless communication markets in an unprecedented way. The increase of these products is a source of capitalization and challenges for the wireless community [1]. Technologies such as Long Term Evolution (LTE) and its successors (LTE-A, LTE-B), strive to provide ubiquitous and high speed cellular connectivity. In order to achieve high throughput in a multi-user context, OFDMA has been chosen as the downlink physical layer access technology. In LTE, OFDMA divides the bandwidth into orthogonal blocks, called Physical Resource Blocks (PRBs) and a frequency domain scheduler assigns such PRBs to the served users based on their channel conditions [2]. Thanks to the Channel State Information (CSI), an LTE base station, can obtain high spectral efficiency by allocating different amounts of PRBs to the users and adapting the transmission modulation and coding schemes (MCS) accordingly. CSI information is hence extremely important for the base station to be able to allocate resources in an efficient way. Although this feedback signalling information (FB) is necessary for the optimal operation of the network, it is also infeasible in an uncompressed way: a full feedback scenario, in which every user transmits CSI to the base station on every PRB at every time instant, would require more than the available LTE uplink bandwidth [3]. Different feedback reduction techniques have been proposed for OFDMA systems, in [4] the authors discuss and compare a number of feedback reduction methods; these can be divided into two categories: *threshold-based* and *Subband grouping* [4]. The first method allows a user to feedback CSI for a PRB only when the channel quality exceeds a pre-determined threshold. While this method reduces the amount of feedback information sent by the users, it does so at the cost of reducing the datarate. The second method allows the users to transmit CSI only on groups of PRBs instead of single ones.

The LTE network also uses the *Subband grouping* feedback reduction, of which 3 different variations are allowed: (1) a wideband feedback, through which users report only one value for the whole bandwidth; (2) a subband level feedback, through which CSI information is reported only for $k$ consecutive PRBs (where $k$ is bandwidth dependent) and (3) a user-selected feedback (also called $Best - M$) through which each user selects the $M$ PRBs which have the best channel and reports CSI valid across the whole selected band [2]. The $Best - M$ policy has been proven to be efficient when paired with opportunistic resource allocation [5] and to reach performance close to the more demanding sub-band level feedback when the number of served users is sufficiently high [6]. In [7] the authors present a heterogeneous FB allocation strategy where the number of FB resources are adapted to the user's channel quality with a maximum throughput scheduler. In [3] the authors show that the number of served users and different fairness strategies, imposed by the frequency resource allocation mechanisms, influence the impact of FB strategies and that determining an optimal FB allocation is possible. The impact of FB information is then a function of the number of users served by the base station, their channel quality and the scheduling algorithm used to assign the PRBs to the users. Both works determine a minimal amount of FB information, valid for all the served users, to obtain the same performances as a full FB system. They do not consider either the practical limitations present in an LTE network and that different FB allocations among users could bring added benefits compared with a single allocation. In this work we first study the impact of FB information on an LTE network already using the pre-compressed FB allocation schemes present in the standard. We show that flexible solutions can provide considerable gain. We propose two reinforcement learning solutions capable of steering the the base station in a position of optimal FB allocation. Both methods make use of Q-Learning [8] to determine which FB allocation maximises the performance and neither require prior training to converge to an optimal solution. The first algorithm finds a single FB allocation valid for all the users while the second, more complex solution, determines a more heterogeneous solution in which the users are allocated different FB amounts based on their channel quality.

The following section describes the LTE system model, it introduces the standard-compliant and newly proposed FB allocation strategies and the scheduling algorithms. In Section III the model used to describe the impact of FB onto the cell's performances is introduced, and Section IV presents the results for various schedulers. Section V discusses the reinforcement learning methods and introduces the concept of Q-Learning. In Section V-B a Q-Learning method to determine the optimal FB allocation across the cell is presented. Section V-C a different Q-Learning method able to assign various FB allocations to different categories of users is presented. In Section VI the simulation results for the two algorithms are discussed. Finally, Section VII draws the concluding remarks.

## II. System Model

### A. Network model

A multi-cell LTE downlink OFDMA scenario is analysed. The network is composed of $B$ base stations (eNodeBs), each serving an equal amount $N_U$ of mobile users (MU). LTE makes use of time-frequency resource allocation: the frequency bandwidth is split into subcarriers grouped into subbands of $N_{sc}$ subcarriers each. The time is slotted and a time slot contains $N_{symb}$ OFDM symbols. Each subcarrier - OFDM symbol pair is named a resource element (RE). The smallest granularity the eNodeB can allocate is composed by $N_{sc} \cdot N_{symb}$ REs and is called a physical resource block (PRB).

In order to allocate resources to the MUs, the base station requests the user's channel quality information on each PRB in the frame. The users measure the signal-to-noise-plus-interference ratio (SINR) for each PRB. The SINR is quantized into a Channel Quality Indicator (CQI) value, indicative of the highest modulation and code rate the base station may use on that PRB while keeping a bit error rate (BER) below a target $10\%$ as shown in table I [9] . Each user then feeds back these CQI values to the base station.

| SINR | CQI | modulation | code rate (x 1024) | efficiency |
|---|---|---|---|---|
| -6.9360 | 1 | QPSK | 78 | 0.1523 |
| -5.1470 | 2 | QPSK | 120 | 0.2344 |
| -3.1800 | 3 | QPSK | 193 | 0.3770 |
| -1.2530 | 4 | QPSK | 308 | 0.6016 |
| 0.7610 | 5 | QPSK | 449 | 0.8770 |
| 2.6990 | 6 | QPSK | 602 | 1.1758 |
| 4.6940 | 7 | 16QAM | 378 | 1.4766 |
| 6.5250 | 8 | 16QAM | 490 | 1.9141 |
| 8.5730 | 9 | 16QAM | 616 | 2.4063 |
| 10.3660 | 10 | 64QAM | 466 | 2.7305 |
| 12.2890 | 11 | 64QAM | 567 | 3.3223 |
| 14.1730 | 12 | 64QAM | 666 | 3.9023 |
| 15.8880 | 13 | 64QAM | 772 | 4.5234 |
| 17.8140 | 14 | 64QAM | 873 | 5.1152 |
| 19.8290 | 15 | 64QAM | 948 | 5.5547 |

TABLE I. SINR AND CQI MAPPING TO MODULATION AND CODING RATE

Once the CQIs for each PRB have been collected, the eNodeB is capable of scheduling resources to each user according to the resource allocation function.

In a practical scenario, however, the CQI reporting is not performed for each PRB, but is quantized in order to reduce the control signalling overhead. The 3 reporting techniques allowed in the LTE standard are presented in the following sub-section [10].

### B. LTE feedback schemes

- Wideband: each user transmits a single 4-bit CQI value for all the PRBs in the bandwidth.
- Higher Layer configured or subband level: the bandwidth is divided into $q$ subbands of $k$ consecutive PRBs and each user feeds back to the base station one 4-bit wideband CQI and a 2-bit differential CQI for each subband. The value of $k$ is bandwidth dependent and is expressed in table II, where $N_{PRB}^{DL}$ is total number of downlink PRBs in the bandwidth.(table 7.2.1-2 in [10]).

| System Bandwidth $N_{PRB}^{DL}$ | Subband Size (k) |
|---|---|
| 6 - 7 | NA |
| 8 - 10 | 4 |
| 11 - 26 | 4 |
| 27 - 63 | 6 |
| 64 - 110 | 8 |

TABLE II. SUBBAND SIZE (K) VS. SYSTEM BANDWIDTH FOR SUBBAND LEVEL FEEDBACK

- User-selected, or $Best - M$: each user selects $M$ preferred subbands of equal size $k$ and will transmit to the base station one 4-bit wideband CQI and a single 2-bit CQI value that reflects the channel quality only over the selected $M$ subbands. Additionally, the user also reports the position of the selected subbands using $P_{FB}$ bits, where $P_{FB}$, as given in [10], is:

$$P_{FB} = \lceil log_2 \binom{N_{PRB}^{DL}}{M} \rceil. \tag{1}$$

The value of $M$ and the amount of PRBs in each subband is given in table III (table 7.2.1-5 in [10]):

| System Bandwidth $N_{PRB}^{DL}$ | Subband Size (k) | M |
|---|---|---|
| 6 - 7 | NA | NA |
| 8 - 10 | 2 | 1 |
| 11 - 26 | 2 | 3 |
| 27 - 63 | 3 | 5 |
| 64 - 110 | 4 | 6 |

TABLE III. SUBBAND SIZE (K) AND NUMBER OF SUBBANDS (M) VS. SYSTEM BANDWIDTH FOR USER-SELECTED FEEDBACK

The three standard compliant feedback schemes do limit the amount of overhead information transmitted by the users, but they do not allow the base station to request a variable amount of feedback to the MU. This could be particularity interesting, firstly, to study the impact that control information has on the data rate and, secondly, to enhance multi-user diversity by allowing different quantities of CQI feedback to users based on their channel conditions. On top of the standard compliant feedback schemes, two extra FB allocation mechanisms have been implemented in order to understand the effects that feedback scarcity has on the downlink capacity;

- Full feedback scheme: each user transmits a 4-bit wideband CQI value and a 2-bit CQI for each PRB. This scheme gives an indication of the maximum capacity the network can achieve when full feedback resolution is available.
- Variable Best-M: This scheme is a flexible implementation of the user-selected one above. The number of subbands $M$ is adapted as a function of the number of users and the system's conditions. Also, there will be a 2-bit CQI value fed back for each subband instead of a single one valid across all subbands. Varying the number of subbands assigned to the users can allow the base station to tailor the amount of FB dynamically, in Section IV the criteria that influence $M$ are analysed.

### C. *Resource Allocation Mechanisms*

While the CQI information defines the rate obtainable on each PRB, the overall cell rate is also function of the resource allocation mechanism implemented in the base station. The scheduling methods, used in this work to define the impact of FB reduction on the cell rate, are described here.

- Best CQI (BCQI) is a greedy scheduler designed to maximise cell throughput. In fact, for each PRB, only the user with the highest channel quality is assigned.
- Proportional Fair (PF): this scheduler is designed to maximise the fraction $\frac{T}{\overline{T}^{\alpha}}$ where $T$ is the instantaneous throughput of the user on PRB, $\overline{T}$ is a moving average of the user's throughput over the previous time slots and $\alpha$ is a fairness coefficient, usually set equal to 1. The PF trades off throughput for fairness.
- Min Max (MM): this method has the objective to maximise fairness at the expenses of throughput.

## III. FEEDBACK MODEL

In this section, we quantify the amount of resources required for the feedback. This model will be complemented with simulation results obtained by optimising feedback and capacity. Table IV includes the bit cost of the different feedback allocation methods presented in section II, where $N_U$ is the number of served users.

| Feedback Scheme | Bit cost |
|---|---|
| Wideband | $2 \cdot (4 \cdot N_U)$ |
| Subband level | $2 \cdot (4 + 2 \cdot q) \cdot N_U$ |
| User-selected | $2 \cdot (4 + 2 + \lceil log_2 \binom{N_{PRB}^{DL}}{M} \rceil) \cdot N_U$ |
| Full feedback | $2 \cdot (4 + 2 \cdot N_{PRB}^{DL}) \cdot N_U$ |
| Variable Best-M | $2 \cdot (4 + 2 \cdot M + \lceil log_2 \binom{N_{PRB}^{DL}}{M} \rceil) \cdot N_U$ |

TABLE IV. SUBBAND SIZE (K) VS. SYSTEM BANDWIDTH

The equations expressed in table IV refer to a single stream of data. If the system makes use of Multiple Input Multiple Output (MIMO), the amount of feedback necessary is multiplied by the number of streams. This does not affect the relation to the network's capacity if UL MIMO is also used for transmitting the feedback information.

Figure 1 shows the amount of feedback required for the different schemes as a function of the number of users with a 20MHz (100 PRBs) UL bandwidth using QPSK modulation. The Figure shows that the full feedback scheme is practically unachievable but the standard compliant methods can use more than 20% of the overall uplink bandwidth for 100 users.

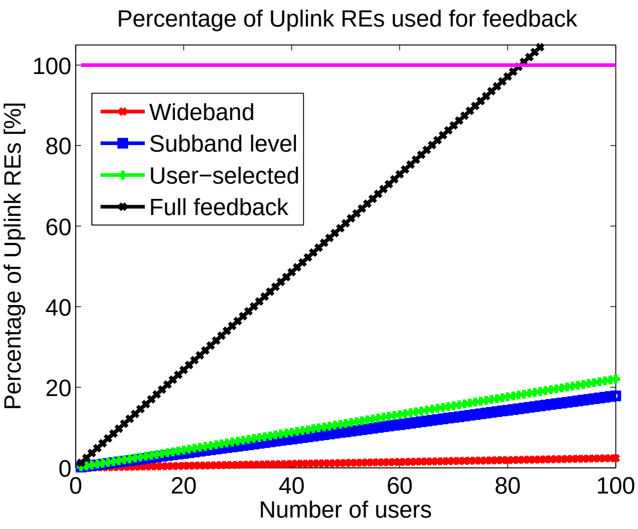


Fig. 1. Portion of Uplink used by FB

As the uplink and downlink PHY channels can carry only a fixed number of modulation symbols but a variable number of bits, in order to quantify the portion of resources that the feedback requires, it is beneficial to redefine the number of feedback bits into modulation symbols.

The uplink bandwidth of the LTE system, although using SC-FDMA instead of OFDMA, also employs PRBs containing an identical number of modulation symbols per physical resource block as the downlink: each PRB can carry 84 modulation symbols and has a duration of 0.5 ms. The symbol rate is then $S = S_{ul} = S_{dl} = 168 \cdot 10^3 \cdot N_{PRB}^{DL}$, in both downlink and uplink [11].

The relation between Baud rate $S$ and the bit rate $T$ is given by:

$$S = \frac{T}{\gamma}, \tag{2}$$

where $\gamma$ is a coefficient which determines how many bits are carried in a symbol; it depends on the modulation and the code rate used. The modulations supported by uplink LTE are QPSK, 16QAM and, only for a highest category of mobile users, 64QAM [11]. Which means that each modulation symbols can carry either 2, 4 or 6 bits for QPSK, 16QAM and 64QAM respectively.

The amount of feedback can then be expressed in modulation symbols, and is here called $S_{fb}$.

The total amount of data transmitted per second is defined as:

$$T_{tot} = T_{dl} + T_{ul} = T_{dl} + T_{ul,data} + T_{ul,fb}, \tag{3}$$

where $T_{dl}$ and $T_{ul}$ represent the throughput in bits in the downlink and the uplink. $T_{ul,data}$ is the amount of payload-only throughput in the uplink and $T_{ul,fb}$ is the feedback throughput obtained by multiplying the bit values expressed in table IV by $10^3$ as each frame is 1 ms long.

Using equation (2) it is possible to define (3) as:

$$T_p = \gamma_{dl} \cdot S_{dl} + \gamma_{ul} \cdot S_{ul} - \gamma_{fb} \cdot S_{ul,fb}, \tag{4}$$

where $T_p = T_{dl} + T_{ul,data}$ is the throughput of the payload data in both uplink and downlink. $\gamma_{ul}$ and $\gamma_{fb}$ are considered, generally, different as the system might request a more robust modulation for signalling information over payload data. Since $S = S_{ul} = S_{dl}$, (4) can be written as:

$$T_p = (\gamma_{dl} + \gamma_{ul}) \cdot S - \gamma_{fb} \cdot S_{ul,fb}. \tag{5}$$

One of the main problems in determining the uplink channel parameters is that uplink and downlink are generally not symmetrical. In case of TDD LTE uplink and downlink bandwidth could be exchanged if more traffic is demanded on one of the two, making the trade-off very relevant. If FDD LTE is used, on the other hand, the downlink and uplink frequency bands are separate. Nevertheless, the feedback information is still reducing the amount of uplink bandwidth available. In order to model the impact of feedback signalling on the uplink performance in a downlink simulator we impose $\gamma_{dl} = 4\gamma_{ul}$, as the LTE downlink spectral efficiency can be up to 4 times higher than the LTE uplink spectral efficiency [11]. Equation (5) becomes then

$$T_p = \frac{5}{4}\gamma_{dl} \cdot S - \gamma_{fb} \cdot S_{ul,fb}. \tag{6}$$

Finally, $\gamma_{dl}$ is obtained directly from (2):

$$T_p = \frac{5}{4}T_{dl} - \gamma_{fb} \cdot S_{ul,fb}. \tag{7}$$

Using equation (7) it is possible to determine whether adding feedback to the system actually improves performance. More feedback would reduce the amount of symbols available for the payload (higher $S_{ul,fb}$) but could also increase the downlink throughput $T_{dl}$. In order to quantify the effect of each user on the total useful throughput $T_p$, it is possible to redefine the total throughput as the sum of the contribution of each user $u$:

$$T_p = \sum_{u=1}^{N_U} T_p^u = \sum_{u=1}^{N_U} \left( \frac{5}{4} T_{dl}^u - \gamma_{fb}^u \cdot S_{ul,fb}^u \right). \tag{8}$$

For the remainder of this paper a value of $\gamma_{fb} = 2$ has been chosen; this is indicative of a 16QAM modulation with a coding rate of 1/2.

## IV. Feedback Impact

In this section we present the impact of feedback compression on the total payload throughput. Specifically, we show how the proposed variable Best-M method allows the base station to request feedback signalling in a more flexible manner and which variables influence the system in order to find the optimal number of subbands $M$.

### A. Simulation Parameters

The system has been simulated using the open source VIENNA system level simulator [12]. An urban multicell environment is considered to include the effects of multipath propagation and interference. Adaptive modulation and coding are used by the base station to allocate resources to the users and there is no cooperation between cells regarding the resource allocation. The simulations are carried out in full buffer in order to account for full bandwidth occupancy and allow for influence of the neighbouring interference on the whole bandwidth. The simulation parameters are included in table V.

| Parameters | Values |
|---|---|
| Number of Macrocells | 7 |
| Sectors per Macrocell | 3 |
| Inter-cell distance | 500 m |
| Macro antenna gain | 15 dB |
| Macro Transmit Power | 46 dBm |
| Macro users per sector | 2 to 100 |
| Frequency | 2.1 GHz |
| System Bandwidth | 20 MHz |
| Number of PRBs | 100 |
| Access technology | OFDMA FDD |
| Number of antennae | 1(Tx and Rx) |
| Channel model | Winner Channel Model II [13] |
| Block fading mean | 0 dB |
| Block fading deviation | 10 dB |
| Fast fading | 10 dB |
| Thermal noise density | -174 dBm/Hz |
| Users speed | 1 m/s |

TABLE V. SYSTEM PARAMETERS

### B. Impact of resource allocation on FB selection

The influence of the different FB allocation strategies for the three schedulers is presented in figures 2 - 4. These figures show the throughput $T_p$ gain of the different strategies over the subband-level allocation for a varying number of served users. The values $1-7$ on the horizontal axis of the figures represent the FB strategy utilised, i.e, 7 indicates the subband-level allocation; 6 represents the user-selected method; and the other values refer to the proposed variable best-M with M varying from 1 to 5.

When the BCQI scheduler is employed, the eNodeB maximises the downlink capacity; the best FB allocation strategy allows the users with best channel quality to obtain the highest throughput. As the cell users increase the impact of FB information becomes increasingly relevant and with 100 concurrent users, the highest number of concurrent transmissions in an LTE cell, choosing the *Variable Best-M* with $M = 1$ FB allocation brings about a 6% gain in total throughput. This limited effect can be ascribed to the BCQI scheduler exploiting multi-user diversity and selecting few users which might contribute to most of the downlink throughput.

The results for the PF scheduler are presented in Figure 3. In this case a much larger gain can be achieved –20% with 100 users– this gain can be attributed to the inherent trade-off between throughput and fairness of the PF scheduler. As the users increase, each individual one gets allocated less PRBs, thus less knowledge of the complete bandwidth is necessary. Furthermore, if a limited amount of PRBs are assigned, the *Variable Best-M* FB allocation allows the base station to have a better information of the users' channel quality only in the portion of bandwidth most likely to be assigned.: i.e. if only 3 PRBs are going to be assigned a *Variable Best-M* with $M = 1$ strategy averages the CQIs over 4 PRBs instead than over 8 like a *subband level* strategy.

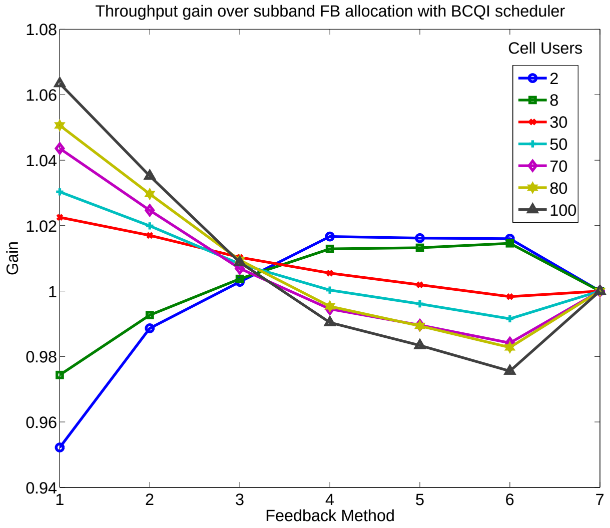


Fig. 2. Throughput gain for BCQI Scheduler for various FB allocation strategies

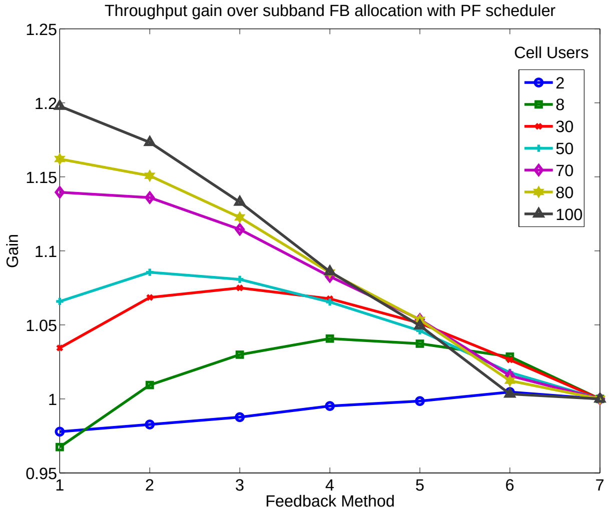


Fig. 3. Throughput gain for PF Scheduler for various FB allocation strategies

Finally, Figure 4 presents the results for the MM scheduler. This scheduler tries to maximise the fairness by improving each user's worst-PRB throughput. Even though this algorithm is the opposite of the BCQI, the impact of FB allocation on the throughput is similar. This is due to the fact that, even though more reliable information is available in the $M$ best subbands fed back by the users, the scheduler is designed to maximise the worst rate and thereby to increase the likelihood that the users are scheduled on a portion of the bandwidth that only reports the wideband CQI. Thus there isn't a real improvement in the downlink rate but the gain comes from not having a loss while reducing FB overhead.

The *Variable Best-M* feedback strategy allows a base station to vary the amount of feedback necessary to maximise payload throughput according the number of served users and scheduling algorithm used by the base station. Table VI presents a

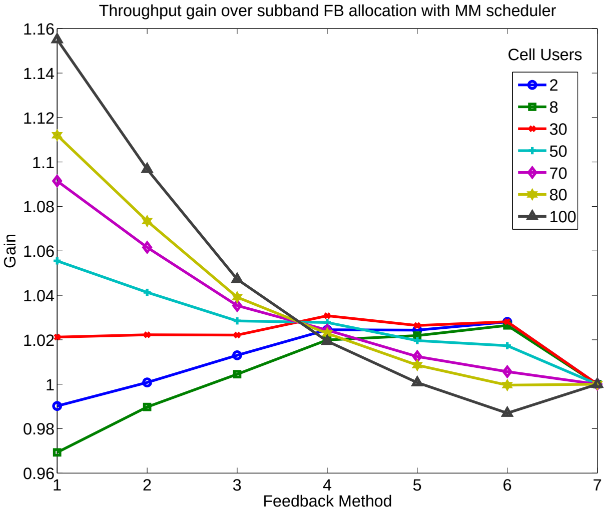


Fig. 4. Throughput gain for MM Scheduler for various FB allocation strategies

compact representation of which homogeneous FB allocation strategy is best to use based on the number of users for each scheduling method.

| Users | Schedulers | | |
|---|---|---|---|
| | Best CQI | Proportional Fair | Max Min |
| 2 | UE-select | UE-select | UE-select |
| 8 | UE-select | Var. Best M, M = 4 | UE-select |
| 30 | Var. Best M, M = 1 | Var. Best M, M = 3 | Var. Best M, M = 4 |
| 50 | Var. Best M, M = 1 | Var. Best M, M = 2 | Var. Best M, M = 1 |
| 70 | Var. Best M, M = 1 | Var. Best M, M = 1 | Var. Best M, M = 1 |
| 80 | Var. Best M, M = 1 | Var. Best M, M = 1 | Var. Best M, M = 1 |
| 100 | Var. Best M, M = 1 | Var. Best M, M = 1 | Var. Best M, M = 1 |

TABLE VI. BEST FB ALLOCATION STRATEGY PER SCHEDULER BASD ON THE NUMBER OF SERVED USERS

If the base station were to allocate the FB dynamically to each user differently, the multi-user diversity could be better exploited. Figure 5 presents a comparison between a homogeneous and a dynamic multi-user FB allocation. The curves in the continuous line represent the best gains for the different number of users of Figures 2 - 4. The curves with dotted line represent the improvement obtainable by performing a dynamic multi-user resource allocation. In order to obtain the results for a dynamic multi-user FB allocation, the simulations have been run at full feedback and the resource allocation decisions of the base stations have been recorded. Afterwards, the simulations have been re-run and only the appropriate amount of FB, computed with the previously obtained results, has been allocated.

As the BCQI is the scheduler that better makes use of multi-user diversity, it is also the one that benefits the most from a dynamic FB allocation. Since the MM scheduler maximises fairness, the amount of PRBs allocated to each user tends to be equal; this way the benefit of a dynamic FB resource allocation is lost. The PF scheduler makes use, albeit in a less extent than the BCQI, of multi-user diversity, and thus sees an improvement with dynamic FB allocation. The gains of the dynamic FB allocation over the static one, in percentage, are presented in table VII. The results for the Max Min scheduler are omitted as the performance improvement was within 0.5% for all users configuration.

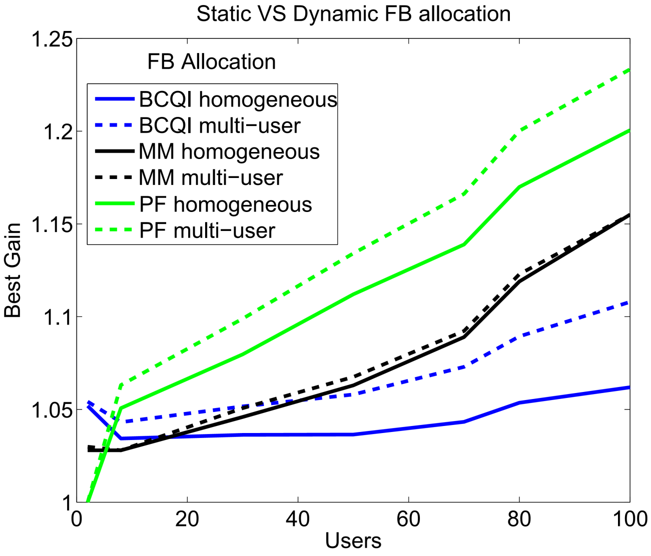


Fig. 5. Gain of dynamic FB VS static FB allocation

| Users | Schedulers | |
|---|---|---|
| | Best CQI | Proportional Fair |
| 2 | 0.25 | 0 |
| 8 | 1 | 1.2 |
| 30 | 1.6 | 2 |
| 50 | 2.2 | 2.2 |
| 70 | 3 | 2.7 |
| 80 | 3.8 | 3 |
| 100 | 5 | 3.3 |

TABLE VII. PERCENTAGE GAIN OF DYNAMIC FB ALLOCATION OVER STATIC ONE FOR BCQI AND PF SCHEDULERS

## V. REINFORCEMENT LEARNING SOLUTIONS

A dynamic controller, capable of positioning the eNodeB in an optimal operating point with respect to the FB allocation, is a desirable device. In the previous section, the proposed variable best-M FB strategy has been proven effective but the system is not able to adjust the number of subbands necessary dynamically with an unknown resource allocation strategy.

Such system has to learn from the cell's current and previous behaviours and makes an informed and intelligent decision on the amount of FB to be allocated to the users. Learning methods have been used successfully in wireless networks and reinforcement learning (RL) is a family of techniques which seems to work particularly well in the context of self-organization and resource allocation problems in LTE [14]–[17].

A great advantage of RL over other learning techniques is its model-free nature. It does not require an extensive representation of the environment and it learns incrementally, without a teacher, until enough information is obtained by the agent. More information on general RL methods is presented in Appendix 1. The following sections will discuss the Q-Learning variant used in this work and the proposed solutions.

### A. Q-Learning Structure

Q-learning is one method of the reinforcement learning family designed to find an optimal action-selection policy by acting upon the environment and determining the impact the action has caused on the current state. By taking and action in a given state, the QL agent learns an action-value function, from which an optimal policy is constructed. Specifically, the QL agent finds a function $Q(s(t), a(t))$ which converges to an optimal value $q^{\pi^*}(s(t), a(t))$ independently of which policy is followed [18].

The system consists in agent and its environment. Each agent can be in any state $s \in \boldsymbol{S}$, can perform any action $a \in \boldsymbol{A}$ to pass from the current state to the next one. Once the action is performed and the new state acquired, a reward $r$ is obtained. The objective of the agent is to maximise the total expected reward and the optimal action, for each state, is the one that presents the highest long term reward. The *learned action-value function* $Q(s(t), a(t))$, also called *Q-value*, is defined as:

$$Q(s(t), a(t)) \leftarrow \\ Q(s(t),a(t)) + \beta \left[ r(t+1) + \gamma(t) \max_a Q(s(t+1),a) - Q(s(t),a(t)) \right], \quad (9)$$

where the *learning factor* $\beta \in [0, 1]$ weights the influence of previous experiences. The smaller the value, the higher is the effect of previous Q-values. $\gamma \in [0, 1]$ is the *discount factor* which limits the influences of future rewards. A high value of $\gamma$ weights greatly the influence that the best action taken for the state $s(t+1)$ has in taking the current action $a(t)$. The system builds, thus, a *Q-Table* of size $\mathbf{S} \cdot \mathbf{A}$ and updates the Q-values at each time interval $t$. All that is required for the convergence of the function is that all state-action pairs are visited [8]; this requirement forces the design of an *exploration-exploitation* policy so that the Q-Table can be completed. In the most common strategy, named $\epsilon - greedy$, the agent chooses an action $a(t+1)$ such as:

$$a(t{+}1) = \begin{cases} \arg\max_a Q(s(t+1), a), & \text{with probability } (1-\epsilon) \\ \textit{random}, & \text{with probability } \epsilon \end{cases} \quad (10)$$

This method allows for continuous exploration with a non-zero $\epsilon$. The value has to be carefully selected so that the systems has enough randomness to visit every action-state pair but is able to exploit the Q-Table so to converge to the optimum. Alternatively, an other frequently used solution assigns high exploration at the beginning of the learning process and gradually diminishes the value, increasing exploitation. A common example of such strategy is given by

$$Pr(a(t+1)) = \frac{e^{\frac{Q(s(t+1),a(t+1))}{\tau}}}{\sum_a e^{\frac{Q(s(t+1),a)}{\tau}}}, \quad (11)$$

where $\tau$ is a temperature function which decays with time [19]. At the beginning the actions are all equally probable and, as time progresses and the Q-Table is built, the actions with highest Q-values will be selected more often. In this work, the problem of determining the correct FB allocation in an LTE network is approached. Since the base station has to decide on the next allocation based on the information fed back by the users, without loss of generality, instead of the transition between times $t$ and $t+1$, the transition $t-1$ to $t$ is evaluated and the action analysed is $a(t-1)$ instead of $a(t)$.

### B. Q-Learning homogeneous FB allocation

For this implementation the learning agent is placed in the eNodeB and has to select a single FB allocation strategy based on the number of served users.

In order to determine the reward for each action, the value of $T_p$, determined in eq (7), is chosen. $T_p$depends on the users' channel quality and on the resource and FB allocation strategies. This means that different channel qualities can give very different results even though the FB allocation strategy remains unchanged. For these reasons the payload throughput is not used as an input state but win the reward function. The state, actions and rewards of the algorithm are here defined.

*1) States:* The state of the base station at time $t$ is defined as

$$\mathbf{S}(t) = \left\{ CQI_{avg}(t), N_{UE}(t) \right\}. \quad (12)$$

Where $CQI_{avg}(t)$ is the average CQI of all the users; this is used in order for the base station to account for channel fluctuations normally occurring in a wireless scenarios and for other effects such as user mobility and interference. A finite number $S_{CQI}$ of quantized CQI states is available for $CQI_{avg}(t)$. $N_{UE}$ is the number of users served by the eNodeB.

*2) Actions:* The set of actions $A$ the agent can take are the different FB methods described in section III; There are then 7 possible actions as shown in table VIII.

| action $a$ | FB allocation |
|---|---|
| 1 | Var. Best M with $M = 1$ |
| 2 | Var. Best M with $M = 2$ |
| 3 | Var. Best M with $M = 3$ |
| 4 | Var. Best M with $M = 4$ |
| 5 | Var. Best M with $M = 5$ |
| 6 | User-select |
| 7 | Subband-select |

TABLE VIII. Possible actions and their relative FB allocation strategies

**Algorithm 1** QL implementation for homogeneous FB allocation

```
Initialization
$t = 0$
$\boldsymbol{QT} \leftarrow \emptyset$
$\boldsymbol{I} \leftarrow \emptyset$
choose random action $a(0)$
for $t$ do
    (1) Receive feedback from the users;
    Evaluate input state:
        $S(t) = \{CQI_{avg}(t), N_{UE}(t)\}$
    (2) eNodeB performs resource allocation with one of the schedulers described in Section II-C .
    (3) Measure the payload throughput $T_p(t)$.
    (4) Update impact matrix $\boldsymbol{I}$ as in (13):
        $I(CQI_{avg}(t), a(t-1), N_{UE}(t)) = \begin{cases} T_p(t), \text{if } T_p(t) > I(CQI_{avg}(t), a(t-1), N_{UE}(t)) \\ I(CQI_{avg}(t), a(t-1), N_{UE}(t)), \text{ otherwise} \end{cases}$
    (5) Compute reward $r(t)$ based on (14)
        $r(t) = \frac{I(CQI_{avg}(t), a(t-1), N_{UE}(t))}{\max I(CQI_{avg}(t), :, N_{UE}(t))};$
    (6) Update the Q-Table $\boldsymbol{QT}$ as in (9):
        $Q(S(t-1), a(t-1)) \leftarrow Q(S(t-1), a(t-1)) + \beta \left[ r(t) + \gamma \max_a Q(S(t), :) - Q(S(t-1), a(t-1)) \right],$
    (7) Choose action $a(t)$ which determines which FB strategy will be used in the next iteration (11):
        $Pr(a(t)) = \frac{e^{\frac{Q(S(t),a(t))}{\tau}}}{\sum_a e^{\frac{Q(S(t),a)}{\tau}}},$
end for
```

*3) Reward:* The throughput $T_p(t)$ determines if the action taken in the previous interval $a(t-1)$ has been beneficial or not. An **impact matrix** $\boldsymbol{I}$ which puts in relation the system's state, the actions and the throughput $T_p(t)$, is used. This matrix has size $S_{CQI} \cdot A \cdot N_{UE}$ and each entry has value:

$$I(CQI_{avg}(t), a(t-1), N_{UE}(t)) = \begin{cases} T_p(t), \text{if } T_p(t) > I(CQI_{avg}(t), a(t-1), N_{UE}(t)) \\ I(CQI_{avg}(t), a(t-1), N_{UE}(t)), \text{ otherwise} \end{cases} \tag{13}$$

The condition that the current value has to be greater than the previous one, in order for the matrix to be updated, is taken from [20], where the authors have shown a greater convergence when this condition is enforced in reinforcement learning.

The reward $r(t)$ is then assigned based on the entries in $\boldsymbol{I}$:

$$r(t) = \frac{I(CQI_{avg}(t), a(t-1), N_{UE}(t))}{\max I(CQI_{avg}(t), :, N_{UE}(t))}, \tag{14}$$

*4) Learning:* The Q-Table **QT** has then the same dimensions as the impact matrix **I** and is updated at every time step $t$ following equation (9). Once **QT** has been updated a new action is selected at instant $t$ based on equation (11) for $t+1$. The algorithmic representation of this implementation is shown in Algorithm 1.

### C. Q-Learning multi-user FB allocation

In case of a dynamic multi-user FB allocation, different users will be able to use different FB methods based on how much they contribute to the system's throughput and how their channel qualities are distributed within the cell. This implementation is build directly from the static one, the structure remains almost unchanged, the major difference is given by the input states which have now to consider, on top of the absolute channel qualities, the data rates of each user and their distribution with respect to each other.

The design of the QL system is explained in the following subsection:

*1) States:* The purpose of the agent is to assign a specific FB allocation method to a user given its channel quality. A new, relative channel quality value $CQI^u_{rel}$, is then introduced to compare the users to each other:

$$CQI^u_{rel} = CQI^u(t) - CQI_{avg}(t), \forall \text{user u} \tag{15}$$

The users are then divided into $N_Q = 5$ categories using the thresholds in table IX.

| Channel Quality | Very Low (VL) | Low (L) | Average (M) | High (H) | Very High (VH) |
|---|---|---|---|---|---|
| $CQI^u_{rel}$ | -5 | -2 | 0 | +2 | +5 |

TABLE IX. CHANNEL QUALITY CATEGORIES AND CQI THRESHOLDS

The state of the base station at time $t$ is then defined as

$$S(t) = \{CQI_{avg}(t), Q_{channel}(t)\}. \tag{16}$$

Where $CQI_{avg}(t)$ is the average CQI of all the users and $Q_{channel}(t)$ indicates whether users of each category are present (e.g. if there are users with channel qualities "Average" and "Very High" then $Q_{channel}(t)$ = [0 0 1 0 1]).

*2) Actions:* The set of available actions is the same as defined in section V-B2. The only difference with the previous implementation is that now $N_Q$ actions are chosen at each time $t$ instead of 1.

*3) Rewards:* Differently than in the single FB allocation algorithm, here the throughput contribution of each user category, $T_Q(t)$ is considered. The value is obtained from $T^u_p(t)$, defined as:

$$T_Q(t) = \frac{1}{N_{U_Q}} \sum_{u^*}^{N_{U_Q}} T^{u^*}_p(t), \forall Q = 1 \cdots N_Q \tag{17}$$

where $N_{U_Q}$ is the number of users belonging to the category $Q$. $T_Q(t)$ represents the throughput contribution of the users in the different quality categories, normalized for one user. The range of these values can vary considerably since it is dependent on the absolute channel quality; it is further impossible to infer if users in a specific category are served

**Algorithm 2** QL implementation for dynamic multi-user FB allocation

```
Initialization
$t = 0$
$\boldsymbol{QT} \leftarrow \emptyset$
$\boldsymbol{I} \leftarrow \emptyset$
choose random actions $a_Q(0)\ \forall Q = 1 : N_Q$
for $t$ do
    (1) Receive feedback from the users and divide them into different channel quality categories; evaluate input state
        $S(t) = \{CQI_{avg}(t), Q_{channel}(t)\}$
    (2) eNodeB performs resource allocation with one of the schedulers described in Section II-C .
    (3) Measure the payload throughput for each category $T_Q(t)$ (17):
        $T_Q(t) = \frac{1}{N_{U_Q}} \sum_{u^*}^{N_{U_Q}} T_{u^*}(t), \forall Q = 1 \cdots N_Q$
    (4) Create categories in which to divide the different channel quality categories based on their throughput contribution (18):
        $RR_Q(t) = \frac{T_Q(t)}{\sum_{q=1}^{N_Q} T_Q(t)};$
    (5) Update impact matrix $\boldsymbol{I}$ for each category as in (19):
        $I(CQI_{avg}(t), Q, a_{Q,t-1})(t) = \begin{cases} RR(t)(T_Q(t)), & \text{if } Q_Q channel(t) \neq 0 \\ 0, & \text{otherwise} \end{cases}$
    (6) Compute each category's reward $\boldsymbol{R}(t)$:
        $r_Q(t) = \boldsymbol{I}(CQI_{avg}(t), Q, a_Q(t-1), \forall Q = 1 \cdots N_Q$
    (7) Update each category's Q-Table $\boldsymbol{QT}$ as in (9):
        $Q(S(t-1), a(t-1)) \leftarrow Q(S(t-1), a(t-1)) + \beta \left[ r(t) + \gamma \max_a Q(S(t), :) - Q(S(t-1), a(t-1)) \right],$
    (8) Choose action $a(t)$ which fixes which FB strategy will be used in the next iteration for each category Q (11):
        $Pr(a(t)) = \frac{e^{\frac{Q(S(t),a(t))}{\tau}}}{\sum_a e^{\frac{Q(S(t),a)}{\tau}}},$
end for
```

consistently more than users in other categories. For example, a user with $CQI^u(t)$ equal to 10 might be in a "Very Good" channel quality group if the average cell CQI $CQI_{avg}(t)$ is 4, but the very same user would have "Low" channel quality if $CQI_{avg}(t)$ were 13. For this reason the contribution of the different channel quality categories to the rate is expressed in relative form:

$$RR_Q(t) = \frac{T_Q(t)}{\sum_{q=1}^{N_Q} T_Q(t)}; \tag{18}$$

At each time $t$, the agent can then build an **impact matrix** $\boldsymbol{I}$, of size $S_{CQI} \cdot N_Q \cdot A$, which relates the input states (average cell CQI and relative user channel quality) with the rate contribution of each category of users and the actions taken. Each entry of $\boldsymbol{I}$ has value:

$$I(t)(CQI_{avg}(t), Q, a_Q(t{-}1)) = \begin{cases} RR_{\hat{Q}}(t), & \text{if } Q_{Q_{channel}}(t) \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{19}$$

Similarly, the reward associated with each action (for every channel quality) $r_Q(t)$ is equal to the same entries of the impact matrix $\boldsymbol{I}$. This way, if users in a specific category are contributing highly (poorly) to the throughput, that category will receive a high (low) reward or will receive no reward if not scheduled.

*4) Learning:* The Q-Table **QT** has then the same dimensions as the impact matrix **I** and is initialized to zero at $t = 0$ QT is updated at every time step $t$ following equation (9). After the update the actions for the following time slot $a_Q(t+1)$ are chosen using equation (11). The algorithmic representation of this implementation is shown in Algorithm 2.

## VI. QL Results

In this section the simulation results for the two proposed algorithms as first presented. The computational complexity and memory requirements of the two methods are then discussed.

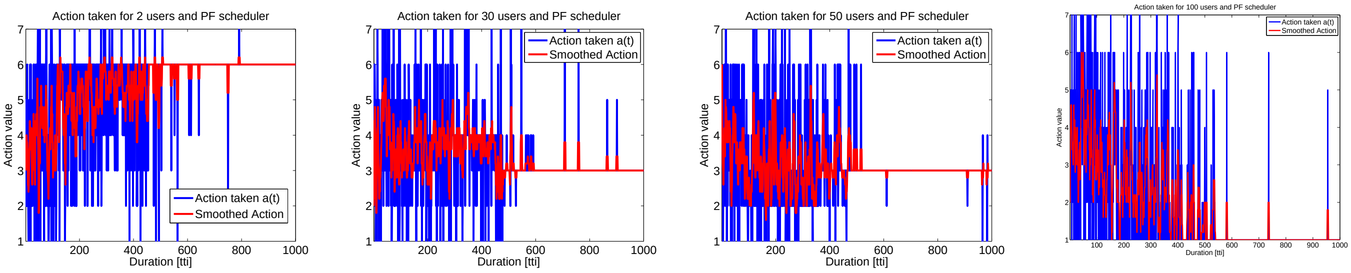
### A. Simulation Results

| | Static solution | Dynamic solution |
|---|---|---|
| State space | $30 \cdot 100$ | $30 \cdot 5$ |
| Action space | 7 | 7 |
| Learning Factor | 0.8 | 0.8 |
| Discount Factor | 0.9 | 0.9 |
| Initial exploration temperature | 200 | 200 |

TABLE X. Learning parameters of the QL algorithms

In this section convergence results for the two proposed Q-Learning algorithms are presented. Simulation settings for the methods are contained in Table X In Figures 6 (a) - (d) the actions taken, at each time interval for a base station using a PF scheduler for samples of 2, 30, 50 and 100 users respectively are shown. This sample of users has been chosen because they require different homogeneous FB allocation actions as shown in Figure 3;

The effective actions taken by the agent are presented in blue, they are selected randomly at the beginning of the simulations. After the initial exploratory phase, each base station converges to the optimal FB allocation determined experimentally in Section IV. This convergence is visible if the action function is smoothed with a moving average filter as the red curve in the figures shows. To further show the convergence and stability of the proposed method to the optimal solutions determined in Section IV-B, the root mean square (RMSE) of the actions taken, with respect to the optimal solutions, is presented in Figure 7. For all the studied user configurations, the proposed solution converges to the optimal static solution and maintains it stably. In case of multi-user FB allocation strategies, without any loss of generalization, only the results for the BCQI scheduler are presented. The agent has to select



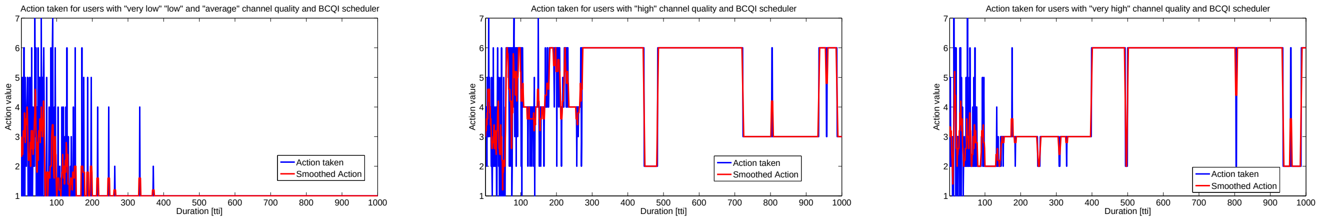
(a)

(b)

(c)

(d)

Fig. 6. Action taken and smoothed action with PF scheduling for 2 (a), 30 (b), 50 (c) and 100 (d) users



(a)

(b)

(c)

Fig. 8. Action taken and smoothed action with a BSQI scheduler for users with for "very low", "low" and "average" channel quality (a), "high" channel quality (b) and "very high" channel quality (c)

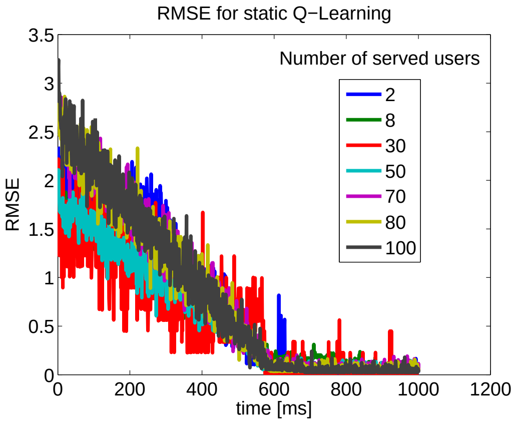


Fig. 7. RMSE convergence for the proposed static QL solution

the best FB allocation based on the channel quality of the users. Figure 8 (a) - (c) shows the action taken, and thus the FB allocation chosen, for users with different channel qualities. Since the BCQI only allocates resources to the users with the best channel quality, the agent learns to allocate only minimal feedback to the users in categories "very low", "low" and "average", while the others get more depending on how good their channel is and how much they contribute to the cell's payload throughput. Users with "very high" channel quality obtain the most feedback. Like in the previous case, the RMSE has been used to verify the convergence and stability of the proposed dynamic solution. Figure 9 shows that the RMSE decreases with each QL iteration and that the final results are very close to the optimal actions. The small oscillations present in the RMSE after convergence is reached are due to the discrete nature of the action-state space. In fact, the proposed method might oscillate between two equally good actions or equally distant from the actual optimal solution.

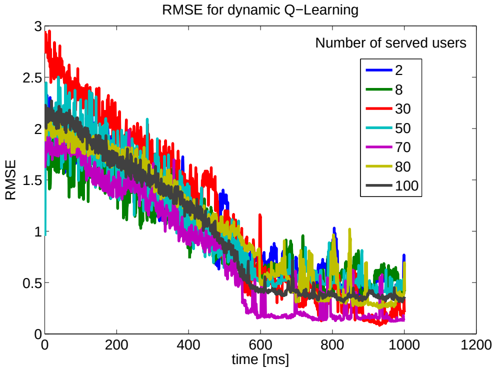


Fig. 9. RMSE convergence for the proposed dynamic QL solution

Finally, in Figure 10 the average results for the QL multi-user FB allocation in case of a BCQI scheduler are compared with the homogeneous allocation of Section IV and the ideal dynamic FB allocation of Figure 5. The proposed multi-user method outperforms the homogeneous allocation and follows asymptotically the ideal solution. The dynamic nature of the multipath propagation environment with mobility users make perfect and reliable allocation very difficult and thus the ideal value is never reached, nonetheless, the proposed solution provides a close to optimal gain (80% of the ideal solution).

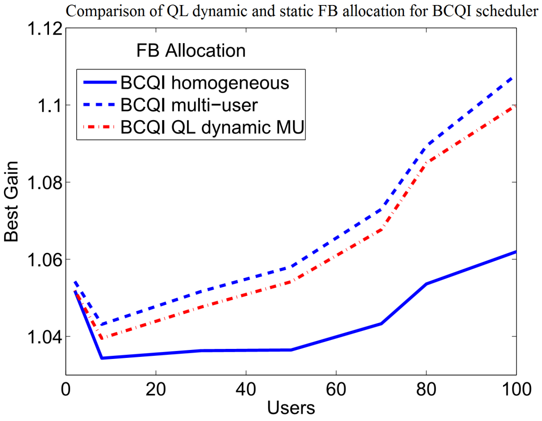


Fig. 10. Comparison for QL dynamic FB allocation with static and ideal dynamic FB allocation

## B. Notes on Complexity

In this section, the complexity of the proposed methods is compared to other operations normally carried out within an LTE base station. It is interesting to note that the proposed solutions make use of information already necessary for the AMC and dynamic frequency scheduling, such as the downlink throughput and the CQI values. This information comes, then, at no extra cost for the eNodeB. In this work, to show the implementation cost of the reinforcement learning methods the memory requirements and the computational complexity are analysed [21]–[23].

*1) Memory Requirements:* The amount of memory of the static and multi-user QL algorithms is directly correlated with the amount of states and actions. The data required is, in fact, contained within the Q-Table and the Impact Matrix, both of dimensions $\boldsymbol{S} \cdot \boldsymbol{A}$, where $\boldsymbol{S}$ represents the number of states and $\boldsymbol{A}$ represents the number of actions. The memory size is then linear in both the number of states and the number of actions: $O(\boldsymbol{SA})$. Specifically, for the static QL algorithm, the number of states is $\boldsymbol{S} = \boldsymbol{S_{CQI}} \cdot \boldsymbol{N_{UE}}$, and the Q-Table has dimensions $\boldsymbol{S} \cdot \boldsymbol{A}$. Given the worst case scenario of $N_{UE} = 100$ and where each entry of the Q-Table is bound to 1 Byte, the total size of the Q-Table is then $(30 \cdot 100 \cdot 7)8 = 168kb$. Thus the total memory space requested by the Q-Table and Impact matrix is $336kb$. Considering instead the dynamic QL algorithm, the Q-Table size is not function of the number of served users but only of the channel quality categories. The number of states is then $\boldsymbol{S} = \boldsymbol{S_{CQI}} \cdot \boldsymbol{N_Q}$. If the same conditions as above are considered, The Q-Table size becomes $(30 \cdot 5 \cdot 7)8 = 8.4kb$. The total memory requirement is then $16.8kb$.

*2) Computational Complexity:* The computational complexity of the QL algorithms is limited by the amount of operations necessary to update the Q-Table. Since at any given moment the agent can be in only one state, the complexity increases linearly with the amount of actions available to the agent [24]. For the problem at hand, the QL agent needs then to determine the current state, update the impact matrix, compute the reward, update the Q-table and finally choose the appropriate action. In a form similar to [23], [25], Tables XI and XII present the total number of operations required by each steps of the static and dynamic solutions respectively. The static method requires only 97 overall instructions per iteration. The dynamic method requires considerably more, 3797 instructions if the absolutely worst case scenario of all 5 categories are present while serving 100 users.

| Steps | Instructions |
|---|---|
| Identification of current and previous states | 2 read<br>30 comparisons |
| Update of impact matrix | 1 read<br>1 comparison<br>1 write |
| Compute reward | 6 read<br>7 comparisons<br>1 division |
| Update Q-Table | 10 read<br>6 comparisons<br>5 MAC<br>1 write |
| Choose next action | 8 read<br>3 divisions<br>7 MAC<br>8 exponentiation |
| Total | 97 |

TABLE XI. COMPUTATIONAL REQUIREMENTS FOR THE STATIC QL METHOD

| Steps | Instructions |
|---|---|
| Identification of current and previous states | 2 read<br>$100 \cdot 30$ comparisons |
| Measure payload for each category | $5 \cdot 100$ MAC<br>5 divisions |
| Create categories | 5 MAC<br>5 division |
| Update of impact matrix | 5 read<br>5 comparisons<br>5 write |
| Compute reward<br>Update Q-Table | 5 read<br>50 read<br>50 comparisons<br>25 MAC<br>5 write |
| Choose next action | 40 read<br>15 divisions<br>35 MAC<br>40 exponentiation |
| Total | 3797 |

TABLE XII. COMPUTATIONAL REQUIREMENTS FOR THE DYNAMIC QL METHOD

The complexity of the proposed methods is actually negligible if compared with other operations normally carried out in an eNodeB base band processor. In fact, at every transmission interval, the base station computes one iteration of the FB reduction methods proposed. At the same time, the base stations has to compute 1 FFT for each of the 14 OFDM symbols present in the frame. For each FFT $2 \cdot Nlog_2(N)$ MAC operations need to be carried out, where $N = 2048$ if the bandwidth is 20 MHz [11]. The total amount of operations is thus 630784. Given the computational requirements of such a necessary operation as the FFT, the impact of the proposed solutions on the processing power of an LTE base station is negligible.

## VII. Conclusion

In this work we show that the feedback overhead cannot be overlooked as the number of connected devices keeps increasing. By using the overall cell throughput model presented in this work, it is possible to identify a trade-off between downlink performance and uplink overhead. Such trade-off is determined by the downlink resource allocation strategy, the number of users served within a cell and their channel quality with respect to the average cell channel quality. It has been shown that, for best CQI, max-min and proportional fair scheduling methods, gains of 11%, 16% and 23% can be expected. The implementation of reinforcement learning solutions can reach almost optimal results in a dynamic environment. The two QL methods presented, one for a static feedback allocation strategy valid for all the users in a cell and the other with dynamic per-user FB allocation, provide very good performance with negligible complexity.

## Appendix
## Reinforcement Learning

Reinforcement learning allows an agent to learn from its environment by acting upon it and observing the effects of such actions. The general structure of RL is depicted in Figure 11.

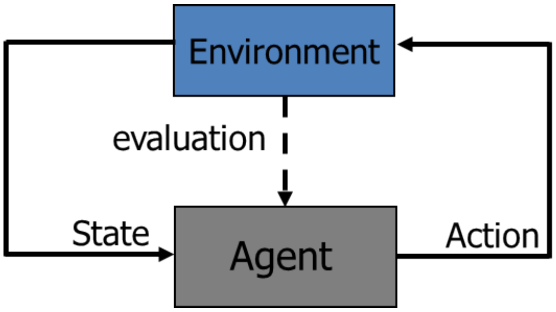


Fig. 11. RL structure

In order for RL to be applied, the system has to be described as a Markov Decision Process (MPD). An MPD is a discrete time stochastic control process useful for systems where the outcome of a decision is partly in control of the agent and partly random. Such process is defined by:

- A discrete number $N_S$ of states $\mathbf{S}$: at each time $t$ the agent monitors the environment via a set of states $\mathbf{S}(t) = s^1(t), s^2(t), s^3(t)...s^{N_S}(t)$.
- A discrete set of actions $\mathbf{A}$: once the condition of the environment is known, the agent performs a different action according to the values of the input states.
- A reward function $\mathbf{R}$: after the actions have been taken, the environment has changed and the states have now shifted from $\mathbf{S}(t)$ to $\mathbf{S}(t+1)$. Associated with this state changes is then a reward $r(t+1)$ indicative of the benefit of such change.
- A state transition function $P^i_{\mathbf{S}(t),\mathbf{S}(t+1)}(a)$ which maps the probability that environment's state will shift from $\mathbf{S}(t)$ to $\mathbf{S}(t+1)$ given that action $a(t)$ is taken at step $t$.

The purpose of RL is to find the optimal policy $\pi^*_s$ that maximises the reward for each state $s$. In the case of *infinite horizon model*, where the lifetime of the agent is unknown a priori, the *value function* that determines which $\pi^*_s$ is the optimal policy is defined as:

$$V^*_{\mathbf{S}(t)} = \max_{\pi} \mathbb{E}\left(\sum_{t=1}^{\infty} \gamma(t) \cdot r(t)\right), \tag{20}$$

where $\pi$ is the complete decision policy and $\gamma(t) \in [0, 1]$ is a discount factor between zero and one which limits the influences of future rewards. $V^*_{\mathbf{S}(t)}$ is then the maximum infinite sum of the discounted rewards that the agent would obtain if it started from state $\mathbf{S}(t)$ and followed policy $\pi^*_{\mathbf{S}(t)}$. Using Bellman's analysis [19] it is possible to determine that such policy exists and that the solution to the value function is unique and given by:

$$V^*_{\mathbf{S}(t)} = \max_{a}\left(r(t+1) + \gamma(t)\sum_{\mathbf{S}(t+1)} P_{\mathbf{S}(t),\mathbf{S}(t+1)}(a(t)) \cdot V^*_{\mathbf{S}(t+1)}\right), \tag{21}$$

The value of the current state $\mathbf{S}(t)$ is then equal to the reward for taking action $a(t)$ summed to the discounted value of the following state when the best action is taken. The optimal policy is then the argument that maximises (21):

$$\pi^*_{\mathbf{S}(t)} = \arg\max_{a}\left(r(t+1) + \gamma(t)\sum_{\mathbf{S}(t+1)} P_{\mathbf{S}(t),\mathbf{S}(t+1)}(a(t)) \cdot V^*_{\mathbf{S}(t+1)}\right). \tag{22}$$

For each policy, the value of taking action $a(t)$ in state $\mathbf{S}(t)$ following policy $\pi_s$ can be determined. The *action-value function* $q_{\pi^*}(\mathbf{S}(t), a(t))$ obtained with the optimal policy $\pi^*_{\mathbf{S}(t)}$, is then defined as:

$$q^{\pi^*}(\mathbf{S}(t), a(t)) = \left(r(t+1) + \gamma(t)\sum_{\mathbf{S}(t+1)} P_{\mathbf{S}(t),\mathbf{S}(t+1)}(a(t)) \cdot V^*_{\mathbf{S}(t+1)}\right). \tag{23}$$

Generally, it is rarely possible to generate optimal policies. The computational and memory costs create the need for approximate solutions such as Q-Learning.

# References

[1] Ericsson White paper, "LTE RELEASE 12," January 2013. [Online]. Available: http://www.ericsson.com/res/docs/whitepapers/wp-lte-release-12.pdf

[2] 3GPP, "UTRA-UTRAN Long Term Evolution (LTE) and 3GPP System Architecture Evolution (SAE)," May 2006.

[3] A. Chiumento, C. Desset, S. Pollin, L. Van der Perre, and R. Lauwereins, "The value of feedback for LTE resource allocation," in *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, April 2014.

[4] S. Guharoy and N. Mehta, "Joint Evaluation of Channel Feedback Schemes, Rate Adaptation, and Scheduling in OFDMA Downlinks With Feedback Delays," *Vehicular Technology, IEEE Transactions on*, vol. 62, no. 4, May 2013.

[5] H. Alyazidi and I. Kostanic, "OFDMA feedback optimization in 4G-LTE systems," in *Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), 2012 IEEE 17th International Workshop on*, Sept 2012, pp. 70–74.

[6] S. Donthi and N. Mehta, "Performance analysis of subband-level channel quality indicator feedback scheme of LTE," pp. 1–5, Jan 2010.

[7] Y. Huang and B. D. Rao, "Performance Analysis of Heterogeneous Feedback Design in an OFDMA Downlink With Partial and Imperfect Feedback." *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 1033–1046, 2013.

[8] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[9] 3GPP TSG-RAN, "3GPP TR 25.814, Physical Layer Aspects for Evolved UTRA (Release 7)," May 2006.

[10] ——, "3GPP TR 36.213, Physical Layer Procedures for Evolved UTRA (Release 10)," July 2012.

[11] S. Sesia, I. Toufik, and M. Baker, *LTE - the UMTS long term evolution : from theory to practice*. Chichester: Wiley, 2009.

[12] J. Ikuno, M. Wrulich, and M. Rupp, "System Level Simulation of LTE Networks," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, may 2010, pp. 1 –5.

[13] P. Kyösti, J. Meinilä, L. Hentilä, X. Zhao, T. Jämsä, C. Schneider, M. Narandzić, M. Milojević, A. Hong, J. Ylitalo, V.-M. Holappa, M. Alatossava, R. Bultitude, Y. de Jong, and T. Rautiainen, "WINNER II Channel Models," EC FP6, Tech. Rep., Sep. 2007. [Online]. Available: http://www.ist-winner.org/deliverables.html

[14] I. Comsa, M. Aydin, S. Zhang, P. Kuonen, and J. F. Wagen, "Reinforcement learning based radio resource scheduling in LTE-advanced," in *Automation and Computing (ICAC), 2011 17th International Conference on*, Sept 2011, pp. 219–224.

[15] M. ul Islam and A. Mitschele-Thiel, "Reinforcement learning strategies for self-organized coverage and capacity optimization," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*, April 2012, pp. 2818–2823.

[16] M. Simsek, M. Bennis, and A. Czylwik, "Dynamic Inter-Cell Interference Coordination in HetNets: A reinforcement learning approach," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, Dec 2012, pp. 5446–5450.

[17] D. Kumar, N. Kanagaraj, and R. Srilakshmi, "Harmonized Q-Learning for radio resource management in LTE based networks," in *ITU Kaleidoscope: Building Sustainable Communities (K-2013), 2013 Proceedings of*, April 2013, pp. 1–8.

[18] C. J. C. H. Watkins, "Learning from delayed rewards," Ph.D. dissertation, King's College, Cambridge, UK, May 1989.

[19] H. Wang and R. Song, "Distributed Q-Learning for Interference Mitigation in Self-Organised Femtocell Networks: Synchronous or Asynchronous?" *Wireless Personal Communications*, vol. 71, no. 4, pp. 2491–2506, 2013. [Online]. Available: http://dx.doi.org/10.1007/s11277-012-0950-6

[20] S. Kapetanakis and D. Kudenko, "Improving on the reinforcement learning of coordination in cooperative multi-agent systems," 2002.

[21] A. L. Strehl, L. Li, E. Wiewiora, J. Langford, and M. L. Littman, "Pac model-free reinforcement learning," in *In: ICML-06: Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 881–888.

[22] M. Jabbarihagh and F. Lahouti, "A decentralized approach to network coding based on learning," in *Information Theory for Wireless Networks, 2007 IEEE Information Theory Workshop on*, July 2007, pp. 1–5.

[23] A. Galindo-Serrano, L. Giupponi, and M. Majoral, "On implementation requirements and performances of Q-Learning for self-organized femtocells," in *GLOBECOM Workshops (GC Wkshps), 2011 IEEE*, Dec 2011, pp. 231–236.

[24] M. Wiering, *Reinforcement learning state-of-the-art*. Berlin New York: Springer, 2012.

[25] A. Galindo-Serrano and L. Giupponi, "Distributed Q-Learning for Aggregated Interference Control in Cognitive Radio Networks," *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 4, pp. 1823–1834, May 2010.