FEDERAL UNIVERSITY OF CEARÁ
DEPARTMENT OF TELEINFORMATICS ENGINEERING
POSTGRADUATE PROGRAM IN TELEINFORMATICS ENGINEERING

MATEUS PONTES MOTA

# REINFORCEMENT LEARNING SOLUTIONS FOR LINK ADAPTATION

FORTALEZA

2020

MATEUS PONTES MOTA

# REINFORCEMENT LEARNING SOLUTIONS FOR LINK ADAPTATION

Presented Thesis for the Post-graduate Program in Teleinformatics Engineering of Federal University of Ceará as a partial requisite to obtain the Ph.D. degree in Teleinformatics Engineering.

Supervisor: Prof. Dr. André Lima Ferrer de Almeida

Co-supervisor: Prof. Dr. Francisco Rodrigo Porto Cavalcanti

FORTALEZA

2020

# Acknowledgements

TODO

# Abstract

TODO

**Keywords:** reinforcement learning, machine learning, link adaptation, rank adaptation.

# Resumo

TODO

**Palavras-chave:** aprendizagem por reforço, aprendizagem de máquina, adaptação de enlace, adaptação de posto.

# List of Figures

2.1 General transmission model on fifth generation (5G) new radio (NR) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12

2.2 Basic diagram of a reinforcement learning (RL) scheme . . . . 14

# List of Tables

# List of Abbreviations and Acronyms

| | |
|---|---|
| 5G | fifth generation |
| AMC | adaptive modulation and coding |
| BCH | broadcast channel |
| CB | code-block |
| CRC | cyclic redundancy check |
| DL-SCH | downlink shared channel |
| eMBB | enhanced mobile broadband |
| FEC | forward error correction |
| HARQ | hybrid automatic repeat request |
| LA | link adaptation |
| LDPC | low density parity check |
| MAC | medium acess control |
| ML | machine learning |
| NR | new radio |
| PCH | paging channel |
| QAM | quadrature amplitude modulation |
| QPSK | quadrature phase shift keying |
| RL | reinforcement learning |
| TTI | transmission time interval |
| UL-SCH | uplink shared channel |

# Table of Contents

1 **Introduction** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

1.1 State-of-the-Art . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

1.1.1 Dual-Connectivity . . . . . . . . . . . . . . . . . . . . . . . 10

1.1.2 Channel Hardening . . . . . . . . . . . . . . . . . . . . . . 10

1.2 Objectives and Thesis Structure . . . . . . . . . . . . . . . . . . . . 10

1.3 Scientific Contributions . . . . . . . . . . . . . . . . . . . . . . . . 10

2 **Conceptual Framework** . . . . . . . . . . . . . . . . . . . . . . . . . . 12

2.1 Transmission Structure . . . . . . . . . . . . . . . . . . . . . . . . 12

2.2 Reinforcement Learning . . . . . . . . . . . . . . . . . . . . . . . . 14

2.2.1 Exploration and Exploitation Trade-off . . . . . . . . . . . . 16

2.2.2 Q-Learning . . . . . . . . . . . . . . . . . . . . . . . . . . 16

3 **adaptive modulation and coding** . . . . . . . . . . . . . . . . . . . . . 18

4 **Link adaptation** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

5 **Conclusions** . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

# Chapter 1

# Introduction

BLA BLA

## 1.1 State-of-the-Art

BLA BLA

### 1.1.1 Dual-Connectivity

BLA BLA

HOHO

### 1.1.2 Channel Hardening

HIHI

## 1.2 Objectives and Thesis Structure

HAHA

## 1.3 Scientific Contributions

Currently, the content of this thesis has been partially published with the following bibliographic information:

***Journal Papers***

- salame

It is worth mentioning that this thesis was developed under the context of Ericsson/UFC technical cooperation projects:

- UFC.40 - *Quality of Service Provision and Control for 5th Generation Wireless Systems*, October/2014 - September/2016;
- UFC.43 - *5G Radio Access Network (5GRAN)*, November/2016 - October/2018,

in which a number of eight technical reports, four in each project, have been delivered. Besides, due to this partnership, two Ph.D. internships took place during this Ph.D.:

- Feb/2016-Jun/2016: Ph.D. internship at Ericsson Research in Luleå-Sweden;
- Sep/2017-Aug/2018: Ph.D. internship at Ericsson Research in Stockholm/Kista - Sweden.

Also in the context of these projects, the author collaborated in the following scientific publication:

**Journal Papers**

- science

Chapter 2

# Conceptual Framework

## 2.1 Transmission Structure

Medium acess control (MAC) uses services from the physical layer in the form of transport channels. A transport channel defines the transmission over the radio interface, by determining its characteristics and how the information is transmitted [1] [2]. The transport channels defined for 5G-NR in the downlink are downlink shared channel (DL-SCH), paging channel (PCH), and broadcast channel (BCH). In the uplink, only one transport-channel is defined, called uplink shared channel (UL-SCH). The data transmissions in the downlink use the DL-SCH and in the uplink the UL-SCH [3]. Data in the transport channel is organized into transport blocks. At each transmission time interval (TTI), up to two transport blocks of dynamic size are delivered to the physical layer and transmitted over the radio interface for each component carrier. [2]

The transmission process is summarized in Figure 2.1. This process is almost the same for the uplink and downlink, the only difference is an additional step of transform precoding after the layer mapping in the uplink case.

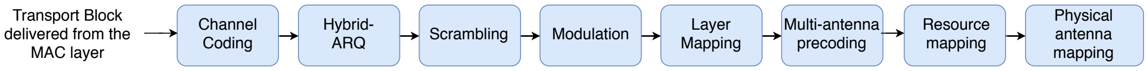


Figure 2.1 – General transmission model on 5G NR

In the modulation phase, NR supports quadrature phase shift keying (QPSK) and three orders of quadrature amplitude modulation (QAM), 16QAM, 64QAM and 256QAM, for both the uplink and downlink, with an additional option of $\pi/2$-BPSK in the uplink. The forward error correction (FEC) code in NR for the enhanced mobile broadband (eMBB) use case in data transmission is the low density parity check (LDPC) code, whereas in the control signaling it is used the polar codes.

The channel coding process in 5G NR is a process composed of six steps [2], namely:

- Cyclic redundancy check (CRC) Attachment: Calculates a CRC and attaches it to each transport block. It facilitates error detection and its size can be of 16 bits or 24 bits.
- Code-block segmentation: Segments the transport block in the case of it being larger in size than the supported by the LDPC coder. code-block (CB) are of equal size.
- Per-CB CRC Attachment: A CRC is calculated and appended to each CB.
- LDPC Encoding: The solution used in NR is a Quasi-cyclic LDPC with two base graphs, the two base matrices that are used to built the different parity-check matrices with different payloads and rates.
- Rate Matching: It adjusts the coding to the allocated resources. Consists of bit selection and bit interleaving.
- Code-Block Concatenation: Concatenates the multiple rate-matching outputs into one block.

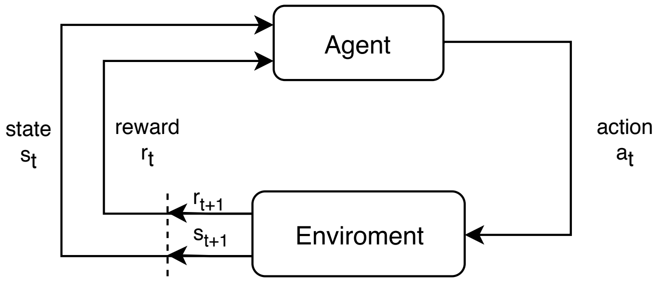
The other blocks in Figure 2.1, excluding the channel coding and the modulation, are:

1. hybrid automatic repeat request (HARQ): 5G NR uses HARQ with soft combining as the primary way to handle retransmissions. In this approach, a buffer is used to store the erroneous packet and this packet is combined with the retransmission to acquire a combined packet, which is more reliable than its components.
2. Scrambling: The process of scrambling is applied to the bits delivered by the HARQ. Scrambling the bits makes them less prone to interference.
3. Layer mapping: The process of layer mapping is applied to the modulated symbols. It distributes the symbols across different transmission layers.
4. Multi-antenna precoding: This step uses a precoder matrix to map the transmission layers to a set of antenna ports.
5. Resource mapping: This process takes the symbols that mapped to be transmitted by each antenna port and these symbols are mapped to the set of available resource elements.
6. Physical antenna mapping: Maps each resource to a physical antenna.

## 2.2 Reinforcement Learning

RL is a machine learning (ML) technique that aims to find the best behavior in a given situation in order to maximize a notion of accumulated reward [4]. Figure 2.2 shows a simple block diagram of the RL problem in which an agent, which is the learner and decision maker, interacts with an environment by taking actions. By its turn, the environment responds to these actions and presents new situations, as states, to the agent [5]. The environment also responds by returning rewards, which the agent tries to maximize by choosing its actions. Unlike supervised learning, where the system learns from examples of optimal outputs, the RL agent learns from trial and error, i.e., from its experience, by interacting with the environment.

Figure 2.2 – Basic diagram of a RL scheme



Source: Created by the author.

At each time step $t$, the agent receives the state of environment $s_t \in \mathcal{S}$, and based on that chooses an action $a_t \in \mathcal{A}$. As consequence of its action, the agent receives a reward $r_{t+1} \in \mathcal{R}$, with $\mathcal{R} \subset \mathbb{R}$, and perceives a new state $s_{t+1}$. In light of this, the basics components of a RL problem are:

- State Space $\mathcal{S}$: Set of all possible states that can be observed by the agent. The random variable $S_t$ denotes the state at time step $t$ and a sample of $S_t$ is denoted $s_t$, with $s_t \in \mathcal{S}$.

- Action Space $\mathcal{A}$: Set of all actions that can be taken by agent. The random variable $A_t$ denotes the action at time step $t$ and a sample of $A_t$ is denoted $a_t$, with $a_t \in \mathcal{A}$

- Transition Probability Space $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0; 1]$ is the transition model of the system, $p(s_{t+1}|s_t, a_t) \in \mathcal{P}$ is the probability of transitioning to state $s_{t+1}$ after taking action $a_t$ in state $s_t$.

- Reward $r_t$: This value indicates the immediate payoff from taking an action $a_t$ in a state $s_t$. $R_t$ is a random variable with a probability distribution depending only of the preceding state and action. We define the expected reward obtained from taking an action $a_t$ in a state $s_t$ as $r(s_t, a_t) = \mathbb{E}\left[R_{t+1} \,|\, S_t = s_t, A_t = a_t\right]$.
- Policy $\pi(s_t) \in \mathcal{A}$: The policy maps the states to actions. More specifically, it maps the perceived states of the environment to the actions to be taken by the agent in those states. The policy can also be defined as $\pi(a_t|s_t)$, the probability of selecting action $a_t$ given the agent is at a state $s_t$.
- Q-function $Q^\pi(s_t, a_t)$: The Q-Function, called action-value function, is the overall expected reward for taking an action $a_t$ in a state $s_t$ and then following a policy $\pi$. It can also be simply denoted as $Q(s_t, a_t)$.

The goal of the RL agent is to find the optimal policy $\pi^*(s_t)$, whose state-action mapping leads to the maximum long term reward given by $G_t = \sum_{t=0}^{\infty} \gamma^t r_{t+1} = r_{t+1} + \gamma G_{t+1}$ [6], where $r_t$ is the received reward at time step $t$. The agent finds its best policy by taking into consideration the value of the Q-function to a state-action pair. Mathematically, the Q-Function is defined as [7]:

$$Q^\pi(s_t, a_t) = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \,|\, S_t = s_t, A_t = a_t\right], s_t \in \mathcal{S}, a_t = \pi(s_t) \in \mathcal{A} \tag{2.1}$$

The parameter $\gamma$ is called *discount factor*, or discount rate, with $0 \leq \gamma \leq 1$. The discount factor is used to control the importance given to future rewards in comparison with immediate rewards, so a reward received $k$ time steps later is worth only $\gamma^{k-1}$ times its value. The infinity sum $\sum_{t=0}^{\infty} \gamma^t r_{t+1}$ has a finite value if $\gamma \leq 1$, as long as the sequence $\{r_k\}$ is bounded [5]. The process is called undiscounted if $\gamma = 1$.

The Q-values in successive steps are related according to the Bellman equation:

$$Q^\pi(s_t, a_t) = \sum_{s_{t+1} \in \mathcal{S}} p\left(s_{t+1} \,|\, s_t, a_t\right) \left[r\left(s_t, a_t\right) + \gamma \sum_{a_{t+1} \in \mathcal{A}} \pi\left(a_{t+1} \,|\, s_{t+1}\right) Q^\pi\left(s_{t+1}, a_{t+1}\right)\right] \tag{2.2}$$

The Equations (2.1) and 2.2 can be rewritten for the case of $\pi$ being the optimal policy. In this case, Equation (2.1) leads to [5]:

$$Q^{\pi^*}\left(s_t, a_t\right) = \mathbb{E}\left[R_{t+1} + \gamma \max_{a_{t+1} \in \mathcal{A}} Q^{\pi^*}\left(S_{t+1}, a_{t+1}\right) \,|\, S_t = s_t, A_t = a_t\right] \tag{2.3}$$

Likewise, assuming the optimal policy, Equation (2.2) leads to [8]:

$$Q^{\pi^*}(s_t, a_t) = r(s_t, a_t) + \gamma \sum_{s_{t+1} \in \mathcal{S}} p\left(s_{t+1} \,|\, s_t, a_t\right) \max_{a_{t+1} \in A} Q^{\pi^*}\left(s_{t+1}, a_{t+1}\right) \tag{2.4}$$

Equation (2.4) can only be solved if we know the transition probabilities. However, if we don't have an adequate model of the environment the agent can take actions and observe their results, then it can fine-tune the policy that decides the best action for each state. The algorithms that explore the environment to find the best policy are called model-free, while those ones that use the transition probabilities are called model-based.

### 2.2.1 Exploration and Exploitation Trade-off

One of the main paradigms in RL is the balancing of exploration and exploitation. The agent is exploiting if is choosing the action that has the greatest estimate of action-value, these are usually called the greedy actions. Whereas exploring is when the agent chooses the non-greedy actions, to improve their estimates. This leads to a better decision-making because of the information the agent has about these non-greedy actions [5].

There are different strategies to control the exploring and exploiting trade off. The reader have a deep discussion on that topic in [9]. In this work, we make use of two strategies:

1. $\epsilon$-greedy: One of the most common exploration strategies. It selects the greedy action with probability $1 - \epsilon$, and a random action with probability $\epsilon$. So, a higher $\epsilon$ means that the agent give more importance to exploration.

2. adaptive $\epsilon$-greedy: There are numerous different methods that adapt the $\epsilon$ over time or as a function of the error [10].A commonly used approach is to start with a high $\epsilon$ and decrease it over time.

### 2.2.2 Q-Learning

In this work, we adopt the Q-learning algorithm, which is an off-policy temporal difference (TD) algorithm. TD methods are model-free and they update their estimates partially based on other estimates, without the need to wait for a final outcome [5]. An off-policy method can learn about the optimal policy at the same time it follows a different policy, called the behavior policy. This behavior policy still has an effect on the algorithm, because it determines the choices of actions. The basic form of the action-values updates is:

$$Q\left(s_t, a_t\right) \leftarrow (1-\alpha) Q\left(s_t, a_t\right) + \alpha\left[r_{t+1} + \gamma \max_{a_{t+1} \in A} Q\left(s_{t+1}, a_{t+1}\right)\right], \tag{2.5}$$

where the parameter $0 \leq \alpha \leq 1$ is called learning rate.

# Chapter 3

# adaptive modulation and coding

AMC

# Chapter 4

# Link adaptation

Link adaptation

# Chapter 5

# Conclusions

Final