國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Department of Electrical Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

有限回授之多天線非正交多工接取系統下使用強化學習選擇調變編碼模式

On Using Reinforcement Learning to Select Modulation/Coding Schemes for Non-Orthogonal Multiple Access in Multi-User Multiple-Input Multiple-Output Systems with Limited Feedback

楊雅涵

Ya-Han Yang

指導教授：謝宏昀 博士

Advisor: Hung-Yun Hsieh, Ph.D.

中華民國 108 年 2 月

February 2019

doi:10.6342/NTU201900453

國立臺灣大學（碩）博士學位論文
口試委員會審定書

有限回授之多天線非正交多工接取系統下使用強化學習選擇調變編碼

On Using Reinforcement Learning to Select Modulation/Coding Schemes for Non-Orthogonal Multiple Access in Multi-User Multiple-Input Multiple-Output Systems with Limited Feedback

本論文係楊雅涵君（R04942076）在國立臺灣大學電信工程學研究所完成之碩（博）士學位論文，於民國 108 年 1 月 17 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

謝志鴻 （簽名）

（指導教授）

（簽名）

所　　長 （簽名）

致謝

碩士時光匆匆而過，收穫遠超出進入碩士前的我能夠想像的。

首先要感謝我的指導老師謝宏昀教授。老師總是能夠適時地指出盲點。「大膽假設，小心求證」，這句從小就耳熟能詳的名言，在碩士生涯有了更深刻的體悟。老師細心的引導，漸漸地培養起我更嚴謹、全面的思考，學到如何有系統地探索新領域，是我一生的收穫。另外也感謝老師讓我有機會去德國交換，和各國家的學生交流，體驗不同的文化。

再來，感謝實驗室夥伴們給的各種溫暖。昀樸和群雄學長，在網管方面提供了許多協助。冠全和閎琛學長不僅常為實驗室帶來歡樂，也總是不吝於分享經驗，畢業後也時常關心大家，是實驗室溫暖的大家長們。Eason 學長在我剛進實驗室的時候給了我許多通訊上的指導，使我更快地上手。昀庭是一路上的好夥伴，在研究上各種幫忙與討論，生活上各種閒聊和照顧，為我的研究生活增添不少樂趣。王鈞是一起準備交換的夥伴，每次聽他分享日本經驗總是很有趣。俊伸的實作能力讓我欽佩，是一開始學習 reinforcement learning 的夥伴。感謝實驗室的學弟妹們，口試時的各種協助，和遠端操作的幫忙。能夠順利口試，學弟妹們功不可沒。我會懷念唱ＫＴＶ的時光，大家都很會 rap，我覺得我快跟不上時代了。

感謝朋友們的鼓勵和支持，很幸運能夠交到一群好朋友，總是願意各種聽我抱怨，嘴砲之餘也不忘和我一起想解決方法。你們像一場及時雨，總是即時的伸出援手，也豐富了生活。相信未來的日子，也能繼續互相扶持。

最後，最感謝的就是家人，總是無條件地支持，使我在求學的路上，沒有後顧之憂，在遇到困難的時候，總是陪伴著我，每次回到家吃完飯，就覺得又是嶄新的一天。未來的人生，換我成為家裡的後盾了。

最後的最後，謝謝那些逝去的人們的關愛，給了我力量去面對挑戰。

doi:10.6342/NTU201900453

摘要

有許多的研究專注於結合多天線多使用者系統（MU-MIMO）和非正交多工接取（NOMA）這兩項技術去提升流量，但這些研究大部分並沒有考慮在實際LTE/LTE-A 環境下，通道資訊回饋是有限的。而根據我們的分析，有限通道資訊回饋造成的量化誤差會導致不合適的資源配置，因此，在結合這兩項技術時，必須考慮如何獲得準確信號與干擾雜訊比（SINR）的方法。為了避免改變目前 LTE-A 的通道回饋規範，本論文採用 Outer Loop Link Adaption（OLLA），利用混合動態回傳（HARQ）來動態調整估 SINR。對於 LTE-A 連結時間短的連線，OLLA 的收斂速度會是一個重要的議題。針對該議題，現行的 OLLA 多只考慮通道品質指標（CQI），但在多天線系統中，預編碼索引（PMI）以及不同排程組合的干擾也須列入考量。在本論文中，我們使用強化學習去改良 OLLA。強化學習可以自動與環境互動，觀察出各種不被現有知識限制的策略。但當將強化學習應用至一個新的領域時，對於該問題的了解會是能否有效訓練的關鍵，因此，我們分析了何種因素會影響 OLLA 的策略。基於該分析，我們考慮了排程過的使用者資訊、通道回饋、偏好的調制與編碼策略和一起排程的使用者以設計合適的特徵擷取、獎賞設計（reward shaping）和探索（exploration and exploitation）機制，並對訓練相關參數進行各種嘗試，以提供更有效率的訓練架構。與 OLLA 的基線相比，我們提出的方法在結合 NOMA+MU-MIMO 時有 7%的增益，在 MU-MIMO 有 14%的增益。此外，OLLA 收斂速度增快了 38%。總而言之，我們提出一個能自動改良 OLLA 的架構，此架構能有效針對不同回饋處理干擾並改善流量和收斂速度。

doi:10.6342/NTU201900453

ABSTRACT

Much work has been done to improve the overall throughput by jointly considering MU-MIMO and NOMA, but little has considered the combination of these techniques under practical environments in LTE/LTE-A, in which the feedback of CSI is limited. Based on our analysis, a method capable of obtaining accurate SINR is important to reduce the improper resource allocation caused by such limited CSI feedback. In this work, to avoid changing the current feedback architecture in LTE-A, we adopt outer loop link adaption (OLLA) to dynamically modify the MCS according to HARQ. Convergence plays a crucial role while applying OLLA in LTE-A due to the characteristics of short connections. For the convergence issue, only CQI is considered in most existing OLLA, while PMI and pairing should be taken into account in MU-MIMO. In this work, we adopt the reinforcement learning to enhance OLLA. Reinforcement learning is a technique which can explore unknown strategies through interacting with the environment. When applying reinforcement learning in a new field, domain knowledge is important for effective training. Therefore, the factors affecting the strategy of OLLA, including the past assigned MCS of the scheduled users, feedback, desired MCS, and pairing user, are analyzed for state design, reward shaping, and exploration and exploitation. Our proposed method improves the throughput by 7% in NOMA+MU-MIMO and by 14% in MU-MIMO. Moreover, the convergence speed is increased by 38%. To conclude, we propose an architecture that can enhance OLLA automatically, deal with the interference, improve the throughput, and accelerate the convergence under different types of feedback.

doi:10.6342/NTU201900453

TABLE OF CONTENTS

doi:10.6342/NTU201900453

doi:10.6342/NTU201900453

LIST OF TABLES

doi:10.6342/NTU201900453

LIST OF FIGURES

doi:10.6342/NTU201900453

doi:10.6342/NTU201900453

doi:10.6342/NTU201900453

CHAPTER 1

INTRODUCTION

With the growth of wireless mobile devices, the increasing demands of wireless mobile connections became an important issue. Thus, multiple access technologies have received a great deal of interest over past years. This technique allows multiple users to share the same wireless medium so the spectral efficiency can be higher.

Toward the trend, the 3rd Generation Partnership Project (3GPP) standardized the radio interface specifications of LTE/LTE-A to enhance the performance for Multiuser Superposition Transmission (MUST). Considerable attention has been paid on Multiple Input Multiple Output (MIMO) and Non-Orthogonal Multiple Access (NOMA). The receiver and the transmitter with multiple antennas, which are called Multiple Input Multiple Output (MIMO), exploit spatial multiplexing, transmit diversity, and beamforming to achieve higher peak rate. Multiuser-MIMO (MU-MIMO) is one of the MIMO techniques allowing multiple users to share the same resource block through beamforming. The term Non-Orthogonal Multiple Access (NOMA) in the thesis indicates the power-domain non-orthogonal multiple access (PD-NOMA), which is also a promising MA technique. It sends multiple messages through different power allocation. The signals transmitted with NOMA can be decoded by the special receiver, which is the so-called successive interference cancellation (SIC). The characteristics of NOMA improve the performance in terms of peak rate as well as fairness. Since both of the techniques have shown their own advantages, and exploit different domains, it is expected naturally that better performance should be seen if combining these two techniques. As a result, we investigate the problems and challenges while combining MU-MIMO with NOMA in the practical LTE/LTE-A environment in this thesis.

Although MU-MIMO can increase the data transmission rate through proper precoding and scheduling, the further improvement is limited in the practical environment due to the limited channel state information (CSI). The quantization error caused by the limited feedback leads to the inaccurate estimation [1], which is one of the major factors of performance loss. [2] discussed the impact of the limited feedback on MU-MIMO. [3] proposed the lower bound of the expectation value of CQI under limited feedback to avoid the overestimation of MCS. This method ensures the reliability but sacrifices the chance of fully utilizing the capacity of

doi:10.6342/NTU201900453

the channel. [4] pointed out that NOMA is capable of increasing both the cell average throughput and fairness. Nevertheless, NONA with SIC also raises the new problem which has never seen in the conventional communication system, such as the pairing of users and the power allocation [5, 6].

It is not until recently that the researches concerning the combination of NOMA and MU-MIMO under perfect CSI are published [7,8]. The combination of the techniques utilizing the power and spatial domain in the same resource block can further improve the spectral efficiency. However, to the best of our knowledge, little investigations have been done in the combination of NOMA and MU-MIMO under practical communication environment. Many studies assumed that the base station can obtain the full knowledge of the channel, while this assumption is impossible in the real world; thus, we would like to further study on the combining NOMA and MU-MIMO in the practical communication environment.

According to our investigation, the quantization error induced by limited feedback is the major factor that the performance in terms of throughput below our expectation in NOMA+MU-MIMO. The quantization error results in the inaccurate estimation of signal to interference plus noise ratio (SINR), so the MCS selection and the power allocation are unable to be decided correctly. Indeed, the approaches to address the scheduling performance loss caused by the CSI impairments have been widely studied. Some paper proposed to dynamically change the link adaption based on the acknowledgment (ACK) and negative acknowledgment (NACK) feedback, known as outer loop link adaption [9, 10]. OLLA aims to deal with the CQI reporting inaccuracy, and compensates for the performance loss to some extent [1]. [11] pointed out that the convergence is a crucial issue for performance when applying OLLA due to the characteristics of short connections in LTE; thus, several studies concerning OLLA focused on increasing the convergence speed. [12, 13] improved the convergence through the analysis of SINR to BLER model. [14] changes the step size based on sequential hypothesis testing and proposed BLER estimator. [15] proposed a method that the step size is based on the elapsed time. Nevertheless, none of these methods are feasible to be applied in differenent scenarios because the performance is highly dependent on the characteristics of the channel model. As the setting of channel model changed, the mathematical model has to be chosen and analyzed again. This will be exhausting work. With the growing complexity of communication environments, it is difficult to find the relationship between the huge number of parameters, and tune the coefficients to handle various communication environments. Thus, the need for a more flexible way to find the relationship between the more complicate parameters is growing. Under these circumstances, the attention on machine

doi:10.6342/NTU201900453

learning based link adaption are rising. Machine learning is known for the capability of capturing the complicated relationship between parameters. [16, 17] have shown that machine learning techniques can capture the complicated effects of the environment to improve performance. In addition, the previous researches regarding convergence in OLLA do not take the convergece speed as objective function, the methods they proposed is based on the observation. Reinforcement learning, which is one of the machine learning techniques, is known for the exploring the unknown strategy through the interaction with the environment while the target is clear. [18–20] using the reinforcement learning to select the MCS.

Although several existing approaches for dynamic link adaption have shown positive results for inaccurate estimation, none of them have suffered from such a severe reporting inaccuracy in the conventional network as much as in NOMA+MU-MIMO due to multiple sources of interferences. There are two interferences in this scenario: one of the interferences is inter-beam interference, which is caused by the MU-MIMO, the projections from the precoding matrix of the other beams can deteriorate the transmission quality. The other is intra-beam interference, which is caused by NOMA, the users with the same precoding matrix but different power allocation induce failed decoding if the paring of users or power allocation is not appropriate. Thus, the deterioration of interference caused by the inaccurate estimation of SINR in the scenario that combining NOMA and MU-MIMO becomes more significant than ever before.

In short, in order to explore the potential of the combination of NOMA and MU-MIMO without ignoring the practical situation in LTE/LTE-A environment and modification of the LTE standard, it is necessary to improve the accuracy of estimated SINR without additional feedback, and to handle the complexity in NOMA+MU-MIMO scenario. In this thesis, we propose the reinforcement learning based dynamic link adaption. In this approach, the selection of MCS is modified based on the acknowledgment (ACK) and negative acknowledgment (NACK) feedback. The optimal strategy of modifying MCS as fast as possible is a crucial problem in the thesis. The optimal strategy is related to the channel response and the past assignment of the MCS of the scheduled users. However, current papers lack discussing the optimal strategy of the MCS selection and exploiting the past knowledge. Thus, we aim to take all these factors into consideration to achieve further performance. As for simulator, we use Vienna [21] as a simulator to simulate the practical LTE-A environments. And, we design the proper reinforcement model to accelerate the training time and improve the convergence speed of finding appropriate SINR estimation. Furthermore, we suggested using SU-MIMO feedback with OLLA according to our investigation. The impact of the

doi:10.6342/NTU201900453

constraint of the retransmissions of the scheduler is also presented in this thesis.

The remainder of the paper is organized as follows: Chapter 2 introduce the background of the current NOMA, MU-MIMO, reinforcement learning, and related work of dynamic link adaption. Chapter 3 describes the system and problem formulation. Chapter 3 analyses the problem. Chapter 4 elaborates the motivation of using reinforcement learning and proposed the reinforcement learning based link adaption. The results are presented in chapter 5. The conclusion is reported in Chapter 6.

doi:10.6342/NTU201900453

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Background

2.1.1 MU-MIMO Overview

MU-MIMO is one of the applications of MIMO. The technique exploits the spatial multiplexing to transmit multiple data streams to multiple users.

As illustrated in Fig. 1, with multiple antennas, the channel responses are various. The transmitted signal, y, after passing the channel response matrix, H, the signal the receiver receives is

$$x = H \times y.$$

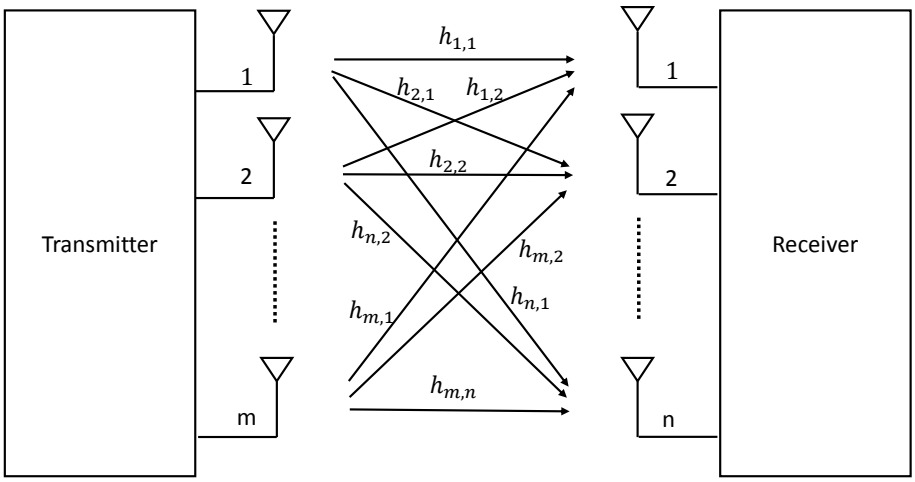


Figure 1: Channel Response for MIMO

It can be seen that the received signal is dependent on the channel response. Thus, many techniques are proposed to utilize to characteristics of the channel for increasing the data rate or the robustness of data transmission.

Beamforming is a signal processing technique utilizing the characteristics of the MIMO. This technique enhances the desired signals and suppresses the interference under a suitable condition. With this technique, the transmitter encodes the signal before transmission. Let $S = [s_1, ..., s_K]$ is the messages that the transmitter

doi:10.6342/NTU201900453

intends to transmit. With beamforming, the transmitted signals are

$$y = F \times S,$$

where F is the precoding matrix.

The mechanism for choosing a precoding matrix is widely researched [22–25]. In general, the assignment of the precoding vector is highly related to the regulation of communication system, such as the capability of the control signal.

In this thesis, we adopt Zero-Forcing beamforming. Let h_k is the channel response of UE_k. $H = [h_1, ..., h_K]$. With Zero-Forcing precoding, the precoding matrix is

$$F = H^H(HH^H)^{-1}. \tag{2.1}$$

Let $F = [f_1, f_2, ..., f_k]$, f_k is the UE_k's precoding vector. Eq. (2.1) implies

$$f_i \times h'_j = 0, \quad \textbf{if } i \neq j. \tag{2.2}$$

That is, if (2.2) holds, the UE_k receives only the signal encoded with w_k, the others will be suppressed.

Based on the standard of the 3GPP E-UTRA long-term evolution (LTE), receivers can get the precoding information from DM-RS (demodulation reference signal) so receivers dont need the explicit precoding information from the transmitter. The method improves the performance while operating MU-MIMO because BS has more freedom in choosing a precoding matrix. Theoretically, as long as the feedback is perfect and the precoding matrix can be chosen arbitrarily, the zero-forcing is able to mitigate the interference. However, in practice, the feedback is limited. In LTE/LTE, the UE returns PMI to indicate the direction of the channel from a codebook $\mathcal{C}$,

$$\mathcal{C} = [c_1, ..., c_C],$$

where c_C is uniform vector. The UE chooses the vector closest to its channel response as its PMI, denoted by $\widehat{h}$.

$$\widehat{h} = arg \max_{c_j \in C} |H_u c_j^*|. \tag{2.3}$$

Under limited feedback, the Zero-Forcing based precoding vector is

$$F = \widehat{H}^H(\widehat{H}\widehat{H}^H)^{-1}, \tag{2.4}$$

where $\widehat{H} = [\widehat{h_1}, ..., \widehat{h_K}]$, $F = [f_1, ..., f_K]$.

Therefore, (2.2) cannot be hold anymore. Let $\widehat{h_k} = \widetilde{h_k} + e_k$, where $\widetilde{h_k}$ is normalized h_k, $\widetilde{h_k} = h_k / \parallel h_k \parallel$.

$$w_i \times \widehat{h}'_j = 0, \quad \textbf{if } i \neq j.$$

doi:10.6342/NTU201900453

$$f_i \times \widetilde{h}'_j = f_i \times (\widehat{h}_k - e_k) = f_i \times e_k. \tag{2.5}$$

It is clear that the message, s_j, encoded with c_j may not be able to mitigate perfectly after passing h_i. As a result, the UE_i may receive the message to others, which are unexpected interference for UE_i. Furthermore, it raises new problems about estimating the accurate SINR for both UEs and base stations.

In this thesis, the method in [3] is adopted. The UEs return the lower bound of SINR based on the assumptions. According to [3], the mean of inner product $|\widetilde{e_k}\widetilde{f_i}|$ is $1/(M-1)$ based on beta-distribution. $|e_k f_k|$ is approximated as 0 due to the assumption that the scheduled UEs are near orthogonal. Following these assumptions, the expectation of $SINR_k$ is

$$\begin{aligned} E[SINR_k] &\geq \frac{\frac{P}{|S|}\left\|h_k\right\|^2 \left|(\widetilde{h_k}\widehat{h}_k^H)(\widehat{h_k}\widetilde{f_k}) + e_k\widetilde{f_k}\right|^2}{1 + \frac{P}{|S|}\left\|h_k\right\|^2 sin^2\theta_k E\left[\sum_{i\in S\setminus k}\left|e_k\widetilde{f_i}\right|^2\right]} \\ &= \frac{\frac{P}{|S|}\left\|h_k\right\|^2 \left|(\widetilde{h_k}\widehat{h}_k^H)(\widehat{h_k}\widetilde{f_k}) + e_k\widetilde{f_k}\right|^2}{1 + \frac{P}{|S|}\frac{|S-1|}{M-1}\left\|h_k\right\|^2 sin^2\theta_k} \\ &\approx \frac{p_k\left\|h_k\right\|^2 cos^2\theta}{1 + \frac{P}{|S|}\frac{|S-1|}{M-1}\left\|h_k\right\|^2 sin^2\theta_k}. \end{aligned} \tag{2.6}$$

The base station uses the lower bound of SINR for scheduling to prevent over-estimation, which may cause failed transmissions. In practice, the UE return

$$g(h_k) = \frac{\frac{P}{M}\left\|h_k\right\|^2 cos^2\theta}{1 + \frac{P}{|M|}\left\|h_k\right\|^2 sin^2\theta_k}. \tag{2.7}$$

The formulation of total lower bound estimated by the base station based on the returned $G(h_k)$ is expressed as

$$G(h_k) = \frac{M}{S\left\|f_k\right\|^2} g(h_k). \tag{2.8}$$

2.1.2 NOMA Overview

NOMA is one of the promising techniques in next communication generation. It meets the increasing demands of wireless mobile connections. Moreover, it improves both overall system throughput and fairness at the same time. The cell-edge user can benefit from NOMA due to the characteristics of NOMA. It is noticing that the gain of cell-edge throughput is improved significantly [6] with NOMA. NOMA is a technique to allocate multiple data on the same resource block(RB), as illustrated in Fig. 2.

doi:10.6342/NTU201900453

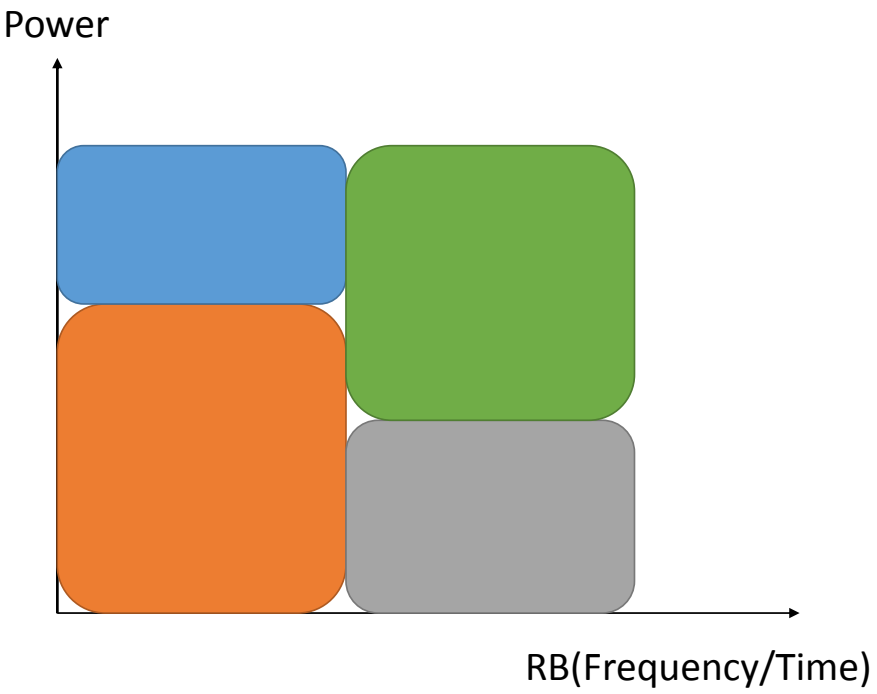


Figure 2: Demonstration of NOMA

The base station use superposition coding to transmit multiple data. The transmitted signal with superposition is,

$$y = a_1 s_1 + ... + a_K s_K, \tag{2.9}$$

where a_k is the power factor of K, $P = \sum_{k \in \mathcal{K}} p_k$; s_k is the signal attempt to transmit to UE_k. The received signal for UE_k is represented as

$$x = h_k(a_1 s_1 + ... + a_K s_K). \tag{2.10}$$

The receivers can use DPC or SIC to decode the signals. In this thesis, we focus on the NOMA with SIC receiver. Assuming there are two users, the decoding process is shown in Fig. 3. The receiver with SIC decodes the stronger but undesired signal iteratively and then substrate the original signal by decoded signal. The procedure does not stop until it can decode its own messages.

Basically, assuming $|h_1| > ... > |h_K|$, the base station allocate power following the criteria [26],

$$a_1 < ... < h_K.$$

The UE_k's throughput is represented as

$$R_k = \sum W log_2(1 + \frac{h_k p_k}{\sum_{k' < k}(h_{k'} p_{k'}) + W N_k}) \tag{2.11}$$

doi:10.6342/NTU201900453

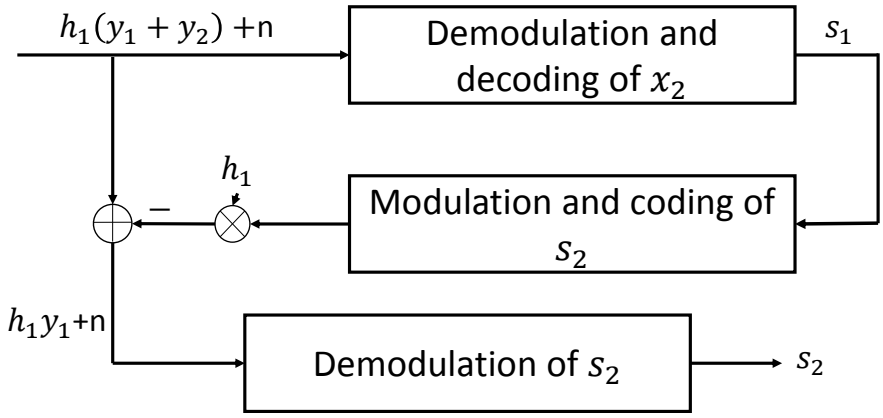


Figure 3: Decoding Process of SIC

In this thesis, we discuss the NOMA with $K = 2$. The user with smaller gain is called far-user while the user with the stronger signal is called near-user. The SIC process only activates in near-user. For far-user, the signal from the other user is too weak to be regarded as interference. Noticing that the success of the decoding of NOMA is highly dependent on the degrading of the signal, it is no surprise the gain difference between users is an important issue while operating NOMA. The impact of the difference between users on overall throughput has been shown in [4]. Assuming that the total power is P, the power of far user is αP, while the power of near user is $(1-\alpha)P$. Therefore, $R_{near}(\alpha)$ and $R_{far}(\alpha)$ are,

$$R_{near}(\alpha) = \sum Wlog_2(1 + \frac{h_{near}\alpha P}{WN_k})$$

$$R_{far}(\alpha) = \sum Wlog_2(1 + \frac{h_{far}(1-\alpha)P}{\alpha P h_{far} + WN_k})$$

Intuitively, the expected total rate $R_{sum}(\alpha) = \sum Wlog_2(1+\frac{h_{near}\alpha P}{WN_k}) + R_{far}(\alpha) = \sum Wlog_2(1+\frac{h_{far}(1-\alpha)P}{\alpha P h_{far}+WN_k})$. The power allocation is an important issue to maximize the throughput [4, 6].

2.1.3 Reinforcement Learning

Reinforcement learning is a learning technique to learn a sequence of actions to achieve better performance. The agent in reinforcement learning learns how to act through the interaction with the environment, as illustrated in Fig. 4. The environment sends observable information, which is called state, to agents. And the agent makes actions in response to the last state and rewards it received.

doi:10.6342/NTU201900453

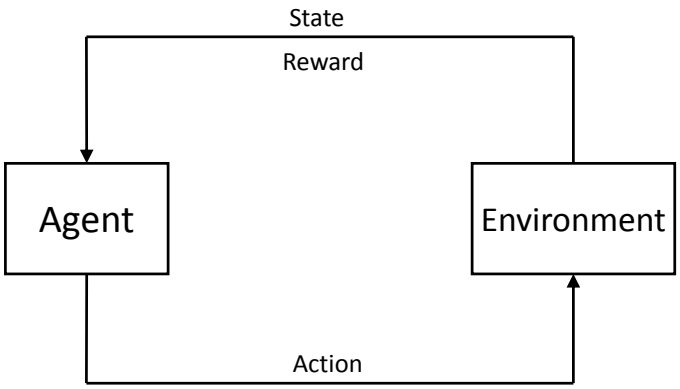


Figure 4: Architecture of Reinforcement Learning

In general, the agent is able to change the state through the interaction with environment.

Unlike supervised learning, which needs the labeled example to learn the correct behaviors, reinforcement learning trains the agent by implicit reward. The agent can learn how good is the action it took instead of the correctness of the actions [27]. The aim of reinforcement learning is to maximize the rewards function in long run. $G_t = R_{t+1} + R_{t+2} + ... + R_T$. Moreover, even the correct behavior is unknown, the agent still can learn the appropriate actions through reinforcement learning. This characteristic of reinforcement learning allows agents to learn without the full knowledge of the problem. It is very useful in many fields since if the agent only learns from the existed knowledge, the agent might never explore the other better behaviors due to the limitation of the known knowledge. Take the game of Go for example [28], machine learning recently makes significant progress in this game, the computer defeats the best player of Go. The contribution of state-of-the-art reinforcement learning plays an important role.

Markov decision process is a crucial elements of the theory and algorithm of reinforcement learning. If a problem can be formulated as a Markov decision process, the method solving such a problem can be regarded as an reinforcement learning methods. Markov decision process satisfies the Markov property. If an environment has the Markov property, the agent is able to make decision based on current state. The property can be represented mathematically as

$$\begin{aligned} p(s', r|s, a) &= Pr(S_{t+1} = s', R_{t+1} = r \mid S_0, A_0, R_1, ...R_t, S_t, A_t) \\ &= Pr(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t). \end{aligned} \tag{2.12}$$

doi:10.6342/NTU201900453

There are four basic elements in Markov decision process listed in the following,

1. $\mathcal{S} = \{s_1, s_2, ..., s_n\}$ denotes the set of n possible states.
2. $\mathcal{A}$ denotes the set of possible actions.
3. $\mathcal{P} = \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ denotes the transition possibility, $p(s, a, s')$, of s to s' while taking the action a
4. $\mathcal{R} = \mathcal{S} \times \mathcal{A} \times \mathcal{S}$ is a reward function. $r(s, a, s')$ express the rewards from s to s'. The expected rewards for the state-action-next-state triples is expressed as

$$\begin{aligned} r(s, a, s') &= \mathbb{E}[R_{t+1} \mid S_t = s, A_t = a, S_{t+1} = s'] \\ &= \frac{\sum_{r \in \Re} rp(s', r \mid s, a))}{p(s' \mid s, a)} \end{aligned} \tag{2.13}$$

Value function $v_\pi(s)$ is a expected return under a policy π in state s. π is a mapping from s, $s \in \mathcal{S}$, to a, $a \in \mathcal{A}$.

Solving a reinforcement learning task is to find a policy that achieves maximal reward in long run. Let $v_*(s) = max_\pi v_\pi(s)1$. $v_*(s)$ can be written as

$$\begin{aligned} v_*(s) &= max_\pi \mathbb{E}_\pi \left[G_t | S_t = s\right] \\ &= max_\pi \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s\right] \\ &= max_\pi \mathbb{E}_\pi \left[R_{t+1} + \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_t = s\right] \\ &= max_\pi \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')], \end{aligned} \tag{2.14}$$

The last equation in (4.2) is the Bellman optimality equation for v_*. Researchers developed many algorithms and methods to solve reinforcement learning tasks. Carefully choosing the proper algorithms and paying attention on the design issue of the algorithm is important to train a agent well.

2.2 Related Work

2.2.1 NOMA+MU-MIMO

NOMA and MU-MIMO techniques can improve the spectral technique. Based on previous works, beamforming, user paring, and power allocation are crucial in order to reach the potential of the techniques. [2] consider the practical feedback system. The cause of the interference in MU-MIMO was well studied. Moreover,

doi:10.6342/NTU201900453

the impact of the limited feedback on MU-MIMO was discussed. It suggests that the CQI should consider interference. [3] proposed the lower bound of the expectation value of CQI under limited feedback to avoid the overestimation of MCS. This way guarantees the reliability but sacrifices the chance of fully utilizing the capacity of the channel. Also, it proposes a scheduler for MU-MIMO. The method in [3] is adopted in VIENNA. [4–6] pointed out that NOMA is capable of increasing both the cell average throughput and fairness. Scheduling constraints and fairness metric can affect the performance.

[8] suggested that the two users can share one precoding matrix. It investigated the impact of the threshold of the correlation between the users on the performance. [7] proposed two precoding technique in order to eliminate the inter-interference. It is noticing that both papers have perfect CSIT assumptions. To the best of our knowledge, little investigations have been done in the combination of NOMA and MU-MIMO under practical communication environment; thus, we would like to further study on the combining NOMA and MU-MIMO in the practical communication environment.

2.2.2 Outer Loop Link Adaption

Outer loop link adaption (OLLA) is a well-known technique to compensate for the inaccuracy of the mapping, CQI imperfection, and the variance of the channel. OLLA solves these problems by modifying the mapping from SINR to MCS dynamically, in contrast to traditional static mapping. OLLA can improve the accuracy of the impractical static mapping for SINR to MCS due to the CQI reporting inaccuracy and the inconsistent channel condition. These inconsistent channel conditions occur because the propagation error condition might be different from the time when constructing the map. The CQI imperfection includes estimation error, which might be caused by the differently calibrated user equipment or hardware inaccuracies and quantization error. The variance of channel condition includes delay of channel reporting(ex: transmission time and decoding time), different numbers of resolvable multi-paths and mobile speeds, and propagation error varies with users.

The concept of OLLA is firstly proposed in [9]. A simple model has developed and analyzed. The method of [9] increases the estimated SNR by a certain fixed step size when receiving an ACK, while the estimated SNR is decreased when receiving NACK. It is noticing that there is a relationship between the step size of increasing SNR and decreasing SNR in order to ensure the BLER. According to the analytical results, it suggested that the step size has a direct impact on the performance and further work have to be done in order to investigate the trade-off

doi:10.6342/NTU201900453

between convergence and power excess.

[29] implemented OLLA in LTE. Although the performance loss is more significant when CQI inaccuracy increasing, it is observed that OLLA is capable of compensating for the performance loss while CQI is inaccurate. [11] showed that the convergence is a crucial issue in LTE due to the characteristics of short connections. Thus, there are several researchers aiming to improve the OLLA mechanism in order to deal with the convergence issue. The approach proposed in [30] improves the convergence by adjusting the initial offset. It showed that the proper initial value can accelerate the convergence speed. The algorithm consists of three stages: filtering, aggregation, and statistical computation. The obvious drawback of the method in this paper is that it has to collect large connection data initially to find a medium value. It may result in performance loss in the beginning.

[12, 13] not only proposed methods to solve the convergence problem but also presented a comprehensive analysis and BLER model. The detailed procedure for analyzing the BLER elaborated how to find a proper mathematical model. Also, with the more detailed model in comparison with the model in [9], the proposed OLLA mechanism is more complicated. They took the average BLER and instantaneous BLER into account. In this way, they had more freedom to adjust the change of steps in response to ACK/ NACK. It is shown that the performance can be improved. However, the mathematical model is specific to a certain channel condition. That is, if the complexity of the communication system increases, the analysis of the mathematical model requires exhausting work.

[14] aims to faster convergence to the target BLER region. It defined three different operating modes to decide the step size. Basically, the closer to the BLER region the estimated BLER is, the larger the step size is. The operating mode is decided by the sequential tests of statistical hypothesis(SHT), which can determine which hypothesis(H) is true with a minimum number of observations and BLER estimator. Although the concept of the dynamically changing the step based on the operating mode is attractive, the proposed BLER estimator might fail to choose the proper operating mode while the selected MCS changes too rapidly. [15] proposed a mechanism to recover fast from the idle to active. The magnitude of compensation is decreased when time passing.

In short, although the convergences issue has been received much attention, how to modify the convergence strategy according to the different channels still remains unknown.

doi:10.6342/NTU201900453

2.2.3 Machine Learning based Link Adaption

Recently, there has been a growing interest in the applications of machine learning. In the communication field, the capability of machine learning to deal with the complicated parameters catch many researchers' eyes.

[16] implemented online AMC with support vector machines to capture the channel effect in real time and found out the proper mapping from SNR to MCS. Unfortunately, this method is not suitable while the mapping is not one-to-one. Also, the training set is still too large to converge fast. [17] proposed a low dimensional feature set to increase the AMC accuracy while operating in MIMO. This research simply adopted k-NN. The method showed good performance. However, this method may suffer from excessive training memory and processing time.

Reinforcement learning is one of the machine learning techniques. It is suitable for a goal-oriented game, training the agent to learn how to act to achieve a higher cumulative reward. Several researches paid attention to this method because one of the advantages of reinforcement learning is that it can train online and save memory in comparison to supervised learning. Reinforcement learning can collect the data in a more efficient way because exploration and exploitation is a widely studied issue in this field [27]. [18, 19] adopted Q-learning and showed better performance in comparison with a supervised learning based method. Although the researches have shown the positive results and the potential, the applied reinforcement learning techniques are not efficient enough. Moreover, these methods did not focus on optimizing the convergence strategy, which is an important issue in a practical environment.

doi:10.6342/NTU201900453

CHAPTER 3

SCENARIO AND PROBLEM FORMULATIONS

In this section, we first introduce the system model in LTE/LTE-A. And then, the encountered problems when implementing NOMA+MU-MIMO in current LTE/LTE-A are presented clearly. In the end, we introduce the problem formulation.

Notations: We use upper-case boldface letters for matrices and lower-case boldface for vectors. The operation $(\cdot)^{-1}$, $(\cdot)^T$ and $(\cdot)^H$ denote the inverse, the transpose and the conjugate transpose of matrix respectively. $\mathbb{E}(\cdot)$ stands for the expectation operator, and $\mathbb{C}$ represent the complex value. $|\mathcal{S}|$ denote the size of set $\mathcal{S}$.

3.1 Network Model of NOMA+MU-MIMO

When the base station uses NOMA technique to transmit signal, they allocate the users with the different power to utilize the power-domain. The receivers adopt SIC to eliminate the interference from the other users. When it comes to MU-MIMO, the based station uses the precoding technique to encode the transmitted signal, exploiting the spatial-domain to superpose multiple users' messages in the same resource block. With the combination of NOMA and MU-MIMO, the base stations transmit the signals with different power allocation and precoders for different receivers. In this thesis, different precoder means different beam. The transmitted signal, which is denoted as y, can be written as

$$y = \sum_{b=1}^{N_b} f_b \sum_{u \in K_b} a_{b,u} s_{b,u}, \tag{3.1}$$

where N_b is the maximum number of beams. $s_{b,u}$ is the desired data which receiver UE_u in beam b desires to receive. Also, to apply the NOMA, the users have to be in the same beam.

In order to retrieve the desired signal, the receiver have to decode the received signal successfully. In the communication environment, the channel responses for different users are various so the received signal for each users are different. The

doi:10.6342/NTU201900453

Table 1: Notation Table

Type	Symbols	Definition
Parameters	N_B	The maximal number of beams
	M	The number of antenna transmitter
	N_r	The number of antenna receiver
	N_c	The number of vectors in the codebook
	P	Power constraint on the transmitted signals
Sets	$\mathcal{C}$	Set of codebook
	$\mathcal{U}$	Set of UEs
	$\mathcal{S}$	Set of scheduled UEs
	$\mathcal{K}_b$	Set of shceduled UEs served in beam b
	$\mathcal{T}$	Set of RBs
	$\mathcal{M}$	Set of MCSs
Variables	x_u	The signal received by UE_u.
	$x_{k,b}$	The signal received by UE_k in beam b.
	y	The transmitted signal
	$s_{b,u}$	The desired signal of UE_u in beam b.
	$H_{k,b}$	The channel matrix of UE_k in beam b.
	$h_{k,b}$	The channel vector of UE_k in beam b.
	$\widehat{h}_{k,b}$	The quantized channel vector returned by UE_k in beam b.
	$\widetilde{h}_{k,b}$	The normalized channel response of UE_k in beam b.
	c_j	The codebook vector j in codebook.
	f_b	The precoding vetor of beam b.
	$a_{b,k}$	The power of UE_k in beam b.
	e_k	The error vector for UE_k.
	$\widetilde{e_k}$	The normalized error vector for UE_k.
	$UE_{b,k}$	The UE_k in beams b.
	γ_{eff}	The effective SINR estimated by BS based on the mapping.
	γ'_{eff}	The effective SINR modified by BS.
	θ	The angle between$\widetilde{h}_{k,b}$ and $\widehat{h}_{k,b}$.

doi:10.6342/NTU201900453

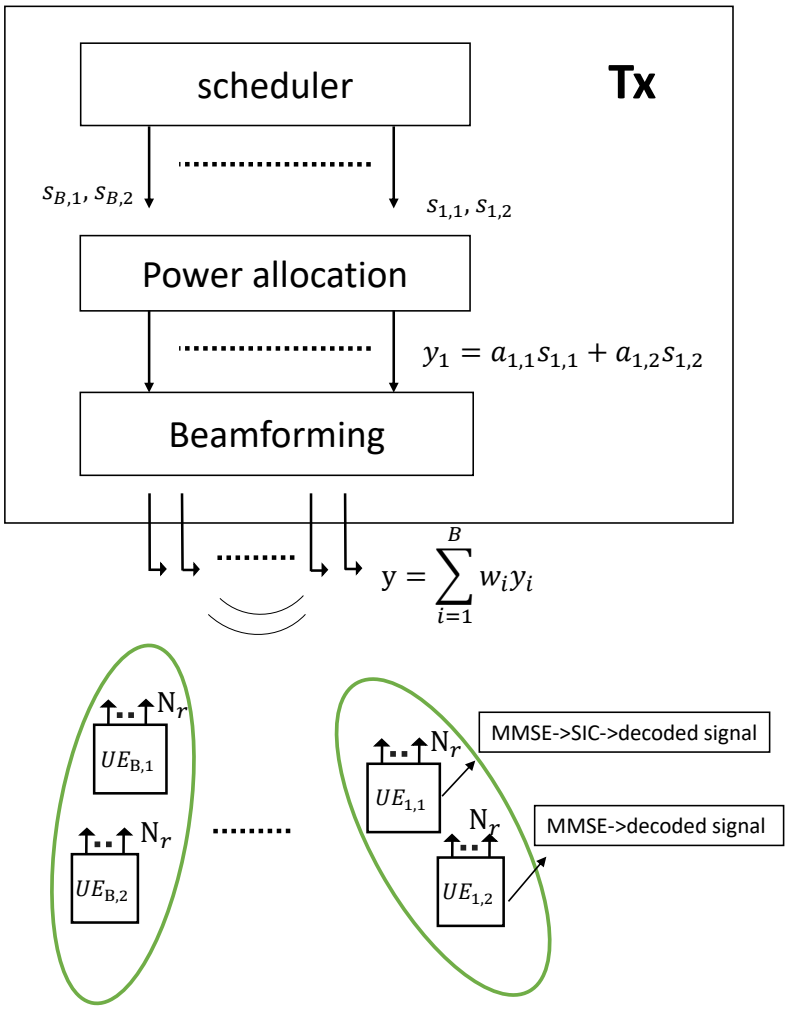


Figure 5: Overall System in NOMA+MU-MIMO

received signal for UE_u's in beam b is denoted by $x_{b,u}$. $x_{b,u}$ is represented as

$$x_{u,b} = H_{u,b} \sum_{b=1}^{N_b} F_b \sum_{u \in K_b} a_{b,u} s_{b,u} + n. \tag{3.2}$$

Also, the user implements spatial filter to enhance the desired signal and suppress the interference including undesired signals.

Thus, the received signal can be rewritten as

$$x = v_{u,b} H_{u,b} \sum_{b=1}^{N_b} F_b \sum_{u \in K_b} a_{b,u} s_{b,u} + n, \tag{3.3}$$

where $v_{u,b}$ is denoted by equalization of receiver.

The effective channel response denoted as $g_{u,b}$ can be represented as

$$g_{u,b} = v_{u,b} H_{u,b}. \tag{3.4}$$

doi:10.6342/NTU201900453

Zero-forcing beamforming is a useful technique to mitigate the interference from the other beam so we implement is as the precoding mechanism. We choose the f_b orthogonal to the other beams. Theoretically, if the channel information can be fully obtained, the received signal with zero-forcing precoding can be rewritten as

$$x = v_{u,b} H_{u,b} F_b \sum_{u \in K_b} a_{b,u} s_{b,u} + n. \tag{3.5}$$

The overall system in NOMA+MU-MIMO is illustrated in Fig. 5.

3.2 Communication System in LTE/ LTE-A

In a practical communication system, the channel information returned by UE is limited. It includes CQI, PMI, and RI. CQI implies the magnitude of the channel. PMI indicates the direction of the channel. RI indicates the number of layers the UE preferred. In LTE/LTE-A, the PMI is chosen from the LTE codebook. Assuming that the number of layers is 1, and the LTE codebook composed of the quantized vector is given by

$$\mathcal{C} = \{c_1, ...c_{N_c}\}, \tag{3.6}$$

where $\mathcal{C}$ denotes the set of the codebook. N_c denotes the number of vectors in the codebook.

The chosen PMI is the best choice among the code book to represent the channel. The chosen PMI, which is denoted by h_u , can be represented as

$$\widehat{h_u} = arg \max_{c_j \in C} |H_u c_j^*|, \tag{3.7}$$

where $\widehat{h_u}$ is best chosen PMI among the codebook.

With zero-forcing beamforming, f_u satisfies

$$f_u \widehat{h}_j^* = 0, \text{if user i is not allocated in the same beam as user j.} \tag{3.8}$$

The received signal of UE u in beam b is

$$x_u = v_{u,b} G_{u,b} f_b \sum_{u \in K_b} a_{b,u} s_{b,u} + v_{u,b} H_{u,b} \sum_{j=1, j \notin b}^{N_b} f_j \sum_{u \in K_j} a_{j,u} s_{j,u} + n. \tag{3.9}$$

The term $\mathfrak{F} = v_{u,b} H_{u,b} \sum_{j=1, j \notin b}^{N_b} f_j \sum_{u \in K_j} a_{j,u} s_{j,u}$ is regarded as undesired signal for UE u. While the perfect CSI is available, $\mathfrak{F}$ would be 0. Otherwise, $\mathfrak{F}$ would be larger than zero.

doi:10.6342/NTU201900453

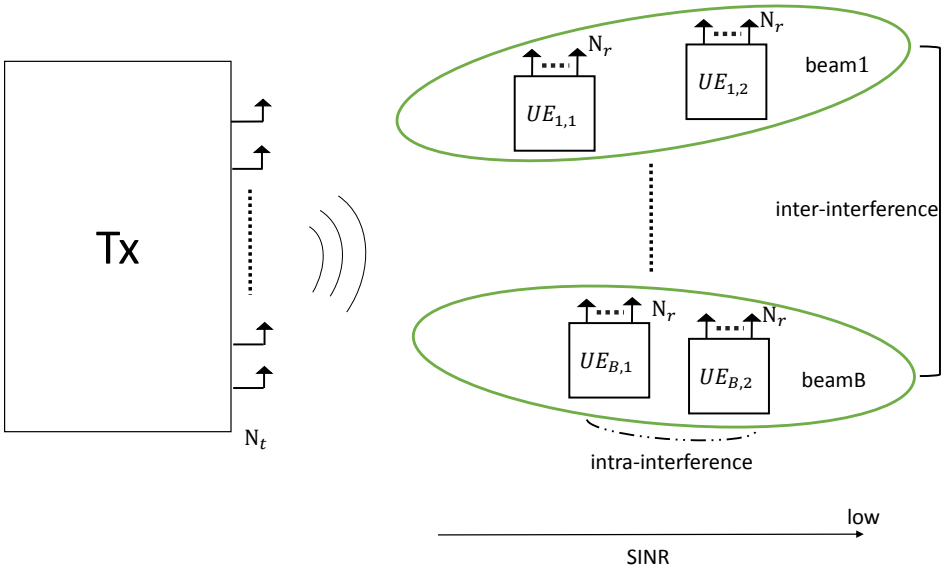


Figure 6: Operate NOMA+MU-MIMO in $N_t \times N_r$ MIMO. Inter-interference indicates the interference caused by the other beams. Intra-interference indicates the interference caused by the other user in the same beam.

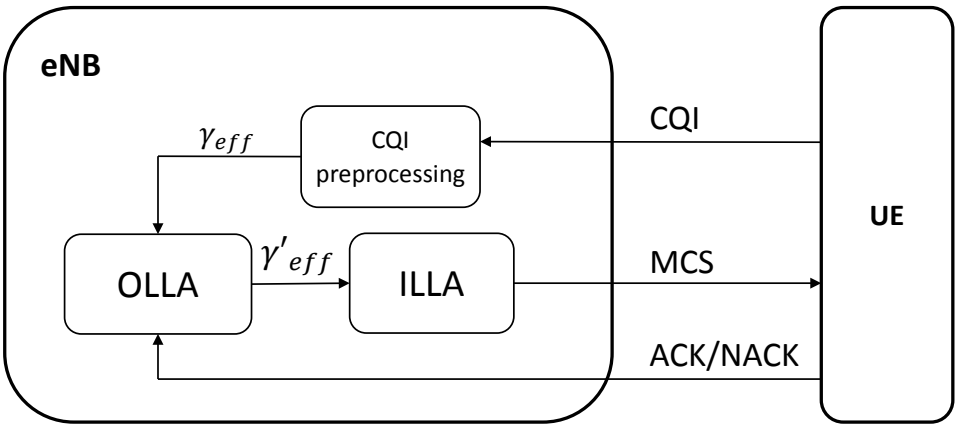


Figure 7: Operation Diagram in Downlink

doi:10.6342/NTU201900453

In Fig. 7, the link adaption scheme is separated into two part: one is the inner loop link adaption (ILLA), the other is outer loop link adaption(OLLA). ILLA is designed to assign the most suitable MCS based on the estimation of link quality. The purpose of OLLA is to correct the reporting inaccuracies. Thus, it modifies the estimated link quality based on the ACK and NACK instead of merely depending on reporting CQI.

3.3 Scenario in LTE/LTE-A

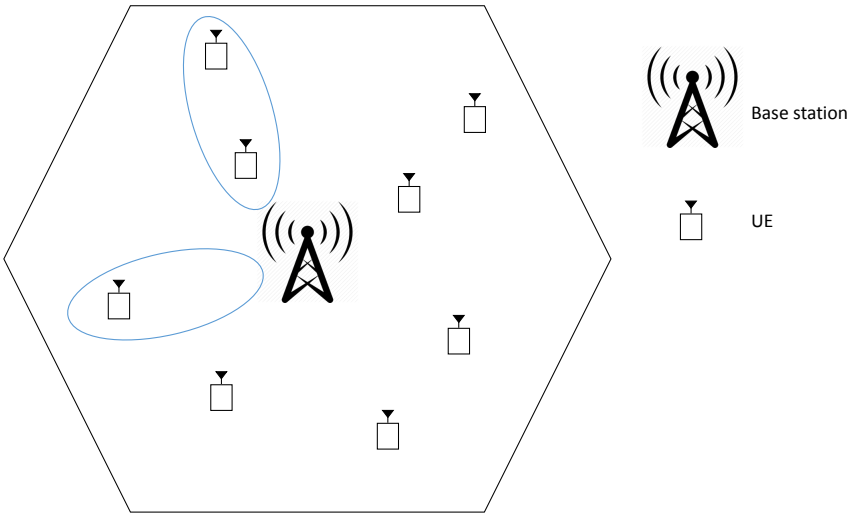


Figure 8: Deployment of NOMA+MU-MIMO

Fig. 8 illustrates the deployment of NOMA+MU-MIMO. The number of transmitters' antenna $N_t = 4$, the number of receivers' antenna $N_r = 1$. The assumptions are shown as following:

1. The number of transmit beams is fixed over all RBs.
2. The power allocated among beams is equal.
3. The number of multiplexed users within a beam is smaller than 2.
4. The maximal number of data streams each user received do not exceed.
5. The PMI is chosen from the LTE codebook.

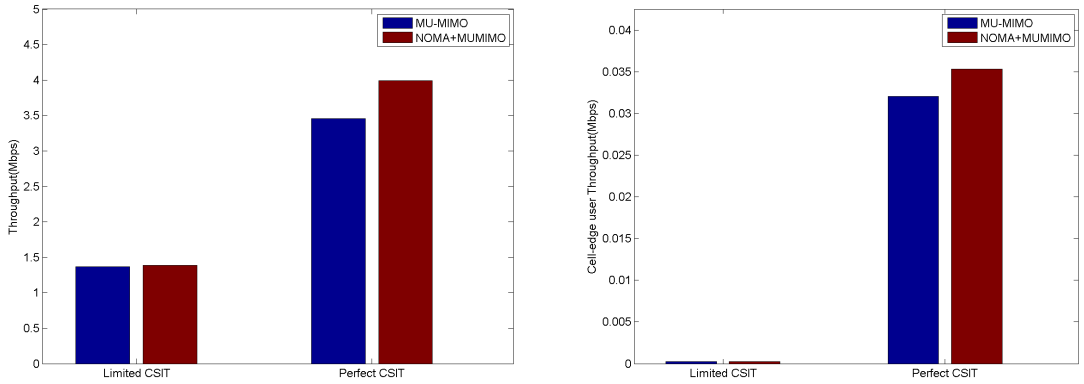
These assumptions allow us to optimize the system performance without losing generality.

doi:10.6342/NTU201900453

3.4 Problems Formulation

3.4.1 Observation of NOMA+MUMIMO and MUMIMO

Firstly, we ran simulations with different the setting of estimation of SINR in order to analyze the the difference between perfect CSI and limited CSI. If the base station can obtain full feedback, it can estimate SINR correctly. By contrast, if the feedback is limited, the base station can only approximate the SINR. The format of limited follows the standard of the LTE\LTE-A. The results are shown in Fig. 9. From Fig. 9, there is almost no throughput gain between NOMA+MUMIMO and MUMIMO. In addition, NOMA+MUMIMO is supposed to be better than MU-MIMO in terms of cell-edge due to the characteristic of NOMA. Unfortunately, the expected results can not be seen from Fig. 9. That is, Fig. 9 indicates that the performance fails to be further improved as long as the estimation is not accurate enough.



(a) Throughput. (b) Cell-edge Throughput.

Figure 9: Comparison between MU-MIMO and NOMA+MU-MIMO with correct Estimation of SINR or not

The previous observations imply that CSI integrity is crucial to the improvement. The feedback adopted in VIENNA is the lower bound of the expectation of SINR, but the inaccuracy of the estimated SINR and real SINR have not analyzed well in [3]. The feedback has an impact on the estimation of SINR so we analyze the variation of real SINR for estimation of SINR based on limited feedback [3].

It is noticing in Fig. 10 that the variation of real SINR for a returned CQI is extremely large. It implies that the returned CQI from UE is very inaccurate. The loss of accuracy of SINR might be acceptable while operating in MU-MIMO. Nevertheless, in the case that taking NOMA into account, the inaccuracy becomes an important issue. Additionally, lots of researches [5,6] have shown that the power allocation and paring is a key problem as operating in NOMA. And the paring mainly depends on the difference of channel gain between the UEs, which receive

doi:10.6342/NTU201900453

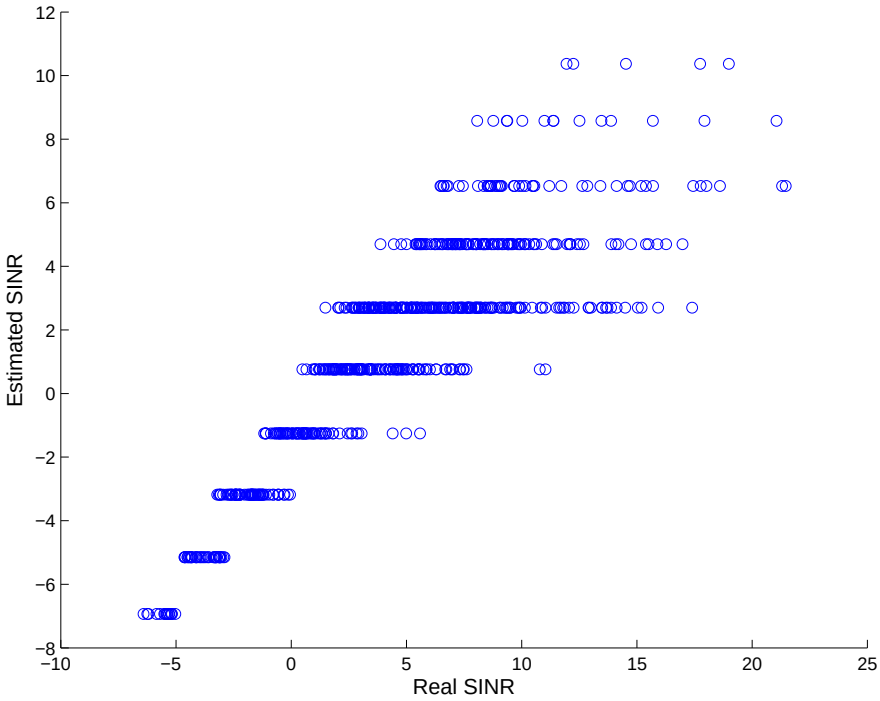


Figure 10: Comparison between estimated SINR and real SINR in terms of Throughput. Estimated SINR is the SINR estimated by limited CSI. Real SINR is SINR that UE actually suffers.

the data encoded with NOMA. The wrong estimation of channel gain difference leads to the wrong power allocation, causing the loss of performance.

Furthermore, the wrong paring and power allocation cause the UE to decode the signal unsuccessfully since the degradation of the signal is more than expected. In other words, the SIC receiver may fail to decode the signal due to the unexpected interference.

3.4.2 The Impact of CSI on SINR

The impact of the inaccurate CSI on SINR will be further discussed.

Assuming that the number of beams is 2, 4x1 MIMO, beamforming is zero-forcing, and the receiver is perfect SIC receiver, which implies that the near user can successfully eliminate the intra-interference. Therefore, the intra-interference can be neglected, the inter-interference is the major topic to be discussed in the following. These assumptions allow us to analyze the problem without losing generality.

If the perfect CSI is available, the relationship between precoder f_i and channel h_i is demonstrated in Fig. 11. After passing the channel, the signal that UE

doi:10.6342/NTU201900453

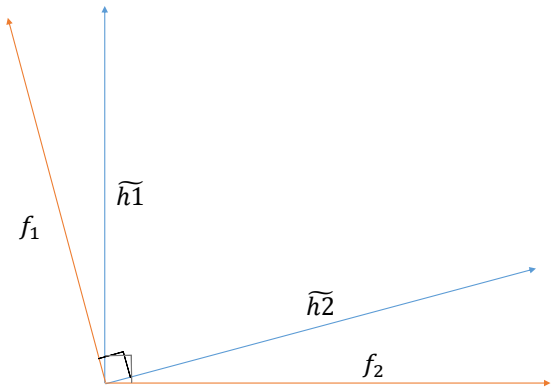


Figure 11: Relationship of f_i and h_i under perfect CSI

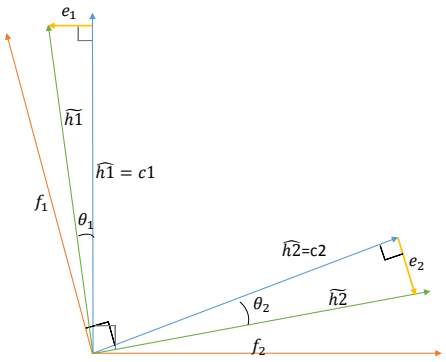


Figure 12: Relationship of f_i and h_i under limited CSI

doi:10.6342/NTU201900453

received is

$$\begin{aligned} Z_1 &= h_1 y \\ &= h_1(f_1 x_1 + w_{2)}) + n \\ &= h_1 f_1 x_1 + h_1 f_2 x_2 + n \\ &= h_1 f_1 x_1 + n \\ &\quad (\because \text{zero-forcing } \therefore h_1 f_2 = 0). \end{aligned} \tag{3.10}$$

Eq. (3.10) shows that the interference from the other beam can be eliminated perfectly with the precoding. However, the situation under the limited feedback is changed.

The relationship of precoder f_i and channel h_i is demonstrated in Fig. 12. The channel vector is $h_k = \|h_k\| \, (|\widetilde{h_k}\widehat{h_k}|^H \widehat{h_k} + e_k)$. e_k is denoted by the error vector. $\widehat{h_k}$ and $\widetilde{h_k}$ are denoted by the quantized channel vector and normalized channel vector, respectively. The received signal of UE is expressed as

$$\begin{aligned} Z_1 &= h_1 y \\ &= \|h_1\| \, (|\widetilde{h_1}\widehat{h_1}|^H \widehat{h_1} + e_1)(f_1 x_1 + f_2 x_2) + n \\ &= \|h_1\| \left((|\widetilde{h_1}\widehat{h_1}|^H \widehat{h_1})(f_1 x_1 + f_2 x_2) + e_1(f_1 x_1 + f_2 x_2) \right) + n \\ &\quad (\because \text{zero-forcing } \therefore |\widetilde{h_1}\widehat{h_1}|^H \widehat{h_1} f_1 = 0) \\ &= \|h_1\| \left((|\widetilde{h_1}\widehat{h_1}|^H \widehat{h_1}) f_1 x_1 + e_1(f_1 x_1 + f_2 x_2) \right) + n \\ &= \|h_1\| \left((|\widetilde{h_1}\widehat{h_1}|^H \widehat{h_1} + e_1) f_1 x_1 + e_1 f_2 x_2 \right) + n. \end{aligned} \tag{3.11}$$

It can be seen that the interference, $\mathfrak{F} = e_1 f_2 x_2$, cannot be eliminated. The limited feedback causes precoding, f_2, to be chosen wrongly because the real channel vector, h_1, is unknown. In this situation, the base station could only choose the precoding orthogonal to $\widehat{h_1}$ instead of h_1. As a result, the messages encoded by the precoding of the other beams cannot be eliminated naturally after going through h_1.

Thus, the SINR of UE_k under limited feedback is

$$SINR_{k,real} = \frac{\frac{P}{|S|} \|h_k\|^2 \left| \widehat{h_k} \widetilde{f_k} cos\theta_k + e_k \widetilde{f_k} \right|^2}{1 + \frac{P}{|S|} \|h_k\|^2 sin^2\theta_k \sum_{i \in S \setminus k} \left| e_k \widetilde{f_i} \right|^2}. \tag{3.12}$$

The interference, $\mathfrak{F} = \sum_{i \in S \setminus k} \left| e_k \widetilde{f_i} \right|^2$, is unknown for UE, since f_i is determined by base station and the co-scheduled UE. Different co-scheduled UEs lead to different $\mathfrak{F}$, which could even range from 0 to 1. On the other hand, the e_k is unknown at the base station due to the quantized PMI. As a result, the base station and UE can not get the accurate estimation of SINR, if the information

doi:10.6342/NTU201900453

Table 2: Objective Function

Objective function:

$$\max_{\pi} \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{T} r_{t+k} \mid s_t \right] \quad (3.13)$$

π is the strategy of selecting MCS.

s_t is the observable information for base station.

$$\mathrm{r}_t = \begin{cases} 0, & \text{if the base station knows that} \\ & \text{it has assigned the suitable MCS .} \\ -1, & \text{otherwise} \end{cases}$$

between the base stations and UE can not be exchanged completely. The base station may fail to estimate SINR for proper scheduling.

3.4.3 Convergence Formulation

Estimation of SINR is highly dependent on the CQI and PMI returned by UE in LTE. On the condition that reporting CSI is unable to represent the real SINR accurate enough due to the quantization error and limited feedback, changing the MCS based on the HARQ information have to be taken into consideration. Changing MCS dynamically based on HARQ is so-called OLLA mechanisim. The prevalence of short connection in LTE network [11] enforces the conventional OLLA to take convergence into consideration. As a result, convergence speed issue is our major goal in the thesis. Mathematically, the objective function is shown in Table 3.4.3.

Eq. (3.13) depicts the convergence problem mathematically.

The larger the Eq. (3.13) is, the quicker the base stations are able to assign the suitable MCS within a period time T. In other words, optimizing the objective function is to fulfill the requirement of the short connections in LTE. The base station can respond to the inaccurate reporting SINR more quickly, achieving better performance in scheduling with the corrected suitable estimation of SINR. Thus, our aim is to design a strategy of modifying the MCS so the base stations can obtain the suitable estimation of SINR as quick as possible.

3.5 Analysis of the Convergence Formulation

To solve the optimization problem, the first step is to analyze the parameters associated with the r_t. r_t is an indicator that whether the base station finds the proper estimated SINR or not. The proper SINR indicates that the base station

doi:10.6342/NTU201900453

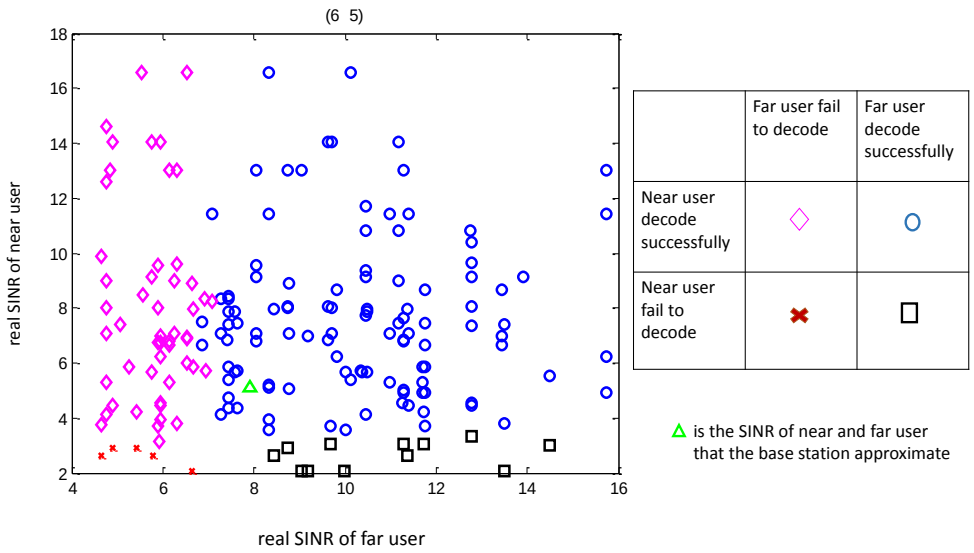


Figure 13: Condition that the receivers of far and near user fail or success to decode the signal

is able to find maximal available MCS for the specified UEs. The base station can judge whether the MCS is maximal available MCS or not by the ACK\NACK. Intuitively, the historical data about the success and failure of the assigned MCS is associated the outcome of the r_t. In addition, we have known the base station chooses the MCS according to the estimated SINR. It decides the MCS based on a map, which suggests the appropriate MCS so the UE can receive the data efficiently, to transform the estimated SINR to MCS. We observed from Fig. 13 that the assigned MCSs only fail on the condition that the estimated SINR is larger than real SINR. That is to say, the probability of the outcome of r_t is associated with not only historical data but also the distribution of SINR. Also, as shown in the previous chapter, the variation of the distribution of SINR is mostly caused by inter-interference. Eq. (3.12) shows all the parameters related to SINR. Therefore, we focus on analyzing the parameters of SINR in terms of PMI, CQI, and co-scheduled PMI.

3.5.1 Analysis of the SINR in MU-MIMO

Assuming the number of the beam is 2, 4x1 MIMO and zero-forcing beamforming. SINR can be written as

$$SINR_{k,real} = \frac{\frac{P}{|S|}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k + e_k\widetilde{f_k}\right|^2}{1+\frac{P}{|S|}\left\|h_k\right\|^2 sin^2\theta_k\sum_{i\in S\backslash k}\left|e_k\widetilde{f_i}\right|^2}. \tag{3.14}$$

Assuming that the co-scheduled UEs are orthogonal, the term $e_k\widetilde{f_k}$ will be 0

doi:10.6342/NTU201900453

with zero-forcing; thus, the real SINR of UE_k can be represented as

$$SINR_{k,real} = \frac{\frac{P}{|S|}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k\right|^2}{1+\frac{P}{|S|}\left\|h_k\right\|^2 sin^2\theta_k\sum_{i\in S\setminus k}\left|e_k\widetilde{f_i}\right|^2}. \tag{3.15}$$

Since the base station can only estimate the SINR by PMI and CQI, we replace the parameters in Eq. (3.15) with PMI and CQI. Through the replacement, the relationship between CSI and SINR is more clear. The distribution of SINR can be observed in a easier way. Knowing that $SINR^2 = |\widehat{h_k}\widetilde{e_l}|^2, cos^2\theta = 1 - sin^2\theta$, the SINR can be rewritten as

$$\begin{aligned}
SINR_{k,real} =& \frac{\frac{P}{|S|}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k\right|^2}{1+\frac{P}{|S|}\left\|h_k\right\|^2 sin^2\theta_k\sum_{i\in S\setminus k}\left|e_k\widetilde{f_i}\right|^2} \\
=& \frac{\frac{P}{|S|}\left\|h_k\right\|^2(1-sin^2\theta_k)\left|\widehat{h_k}\widehat{h_k}^H\right|^2}{1+\frac{P}{|S|}\left\|h_k\right\|^2 sin^2\theta_k\sum_{i\in S\setminus k}\left|e_k\widetilde{f_i}\right|^2} \\
& (\because cos^2\theta = 1 - sin^2\theta) \\
=& \frac{\frac{P}{|S|}\left\|h_k\right\|^2(1-|\widehat{h_k}\widetilde{e_k}|^2)}{1+\frac{P}{|S|}\left\|h_k\right\|^2|\widehat{h_k}\widetilde{e_k}|^2\sum_{i\in S\setminus k}\left|e_k\widetilde{f_i}\right|^2} \\
& (\because sin^2\theta = |\widehat{h_k}\widetilde{e_k}|^2) \\
\propto& \frac{\frac{P}{|S|}\left\|CQI_k\right\|^2(1-|PMI_k\widetilde{e_k}|^2)}{1+\frac{P}{|S|}\left\|CQI_k\right\|^2|PMI_k\widetilde{e_k}|^2\sum_{i\in S\setminus k}|e_kPMI_i|^2} \\
& (\because CQI_k \propto CQI_k, \text{letting } PMI_k \text{ indicate } \widehat{h_k})
\end{aligned} \tag{3.16}$$

We can conclude from Eq. (3.16) that the e_k is a random variable, which is related to PMI_k and CQI_k. The inner product of e_k and PMI_i can affect the $SINR_k$. Also, it can be observed that the interference is caused by $f_k\widetilde{e_k}$ and e_kf_i. The f_i and f_k are precoding vectors for UE_i and UE_k respectively. That is to say, if the UE knows its co-scheduled user in advance, it could return the perfect estimation of CQI; or, if the base station is able to get the information of e_k, the perfect estimation of SINR can be achieved. Nevertheless, the additional feedback would bring more burden on the channel because the channel has to give more space for transmitting control signal instead of the data. It is a trade-off

doi:10.6342/NTU201900453

between the bits of feedback and the accuracy of the estimation. Furthermore, the additional feedback has to change the current LTE feedback standard. The target of this thesis is to assign a suitable MCS in order to achieve a better performance following the LTE feedback standard. Thus, the target of analyzing the SINR is to improve the design of the mechanism to find the proper MCS dynamically. As a result, we focus on the analysis of random variable in Eq. (3.16) for the reason that the probability of whether the transmission is failed or successful is highly associated with these parameters.

3.5.2 Observation of SINR in different CQI and PMI

In the previous subsection, we learned that the distribution of SINR is associated with the PMI, CQI, and co-scheduled CQI mathematically, but the actual distributions are still unknown. As a result, we run the simulations for different CQI and PMI to observe the pattern of the distribution.

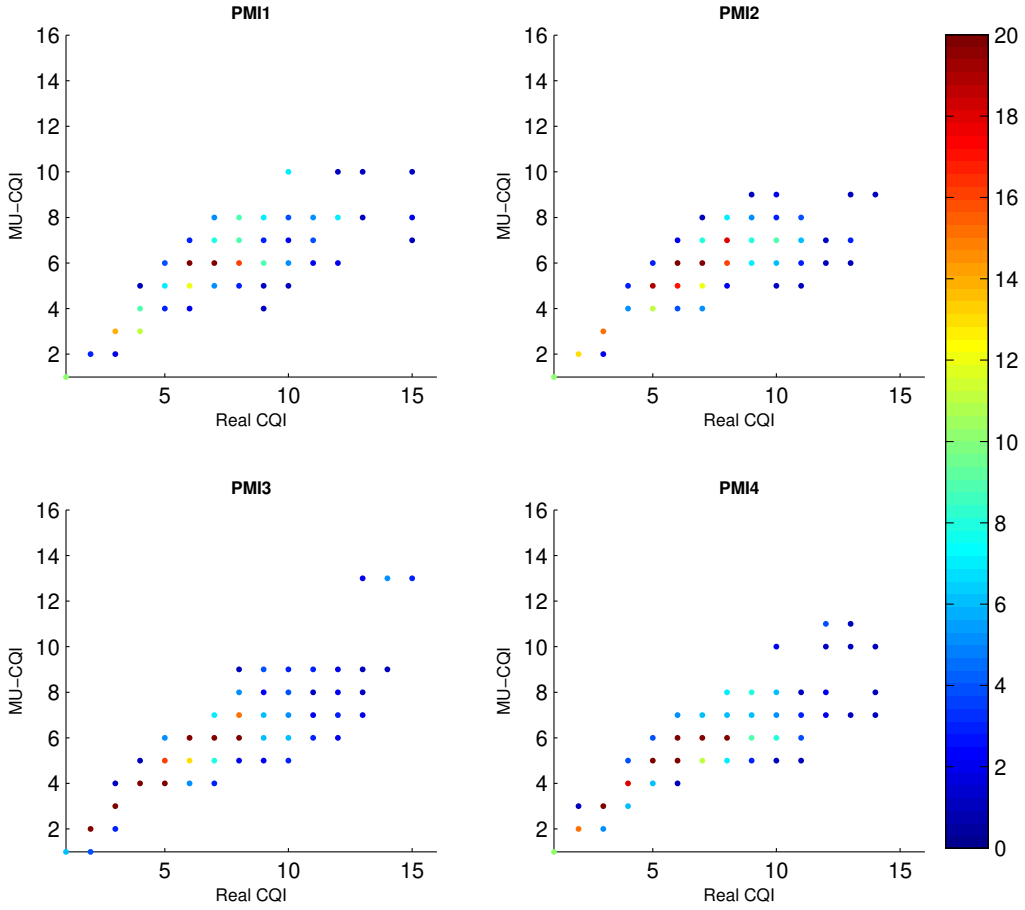


Figure 14: Real CQI and estimated CQI. Real CQI is calculated by h, the estimated CQI is the CQI returned by UE.

doi:10.6342/NTU201900453

In Fig. 14, only four different PMI cases are presented for the convenience of observation. The x-axis is the real CQI value calculated based on real channel response. The y-axis is the estimated CQI value calculated based on limited feedback. The color represents the number of the pair of certain estimated CQI and real CQI out of the overall cases. Basically, if the point tends to be red, it implies such a pair appears more frequently. This figure indicates that what is the possible real value of CQI corresponding to the estimated value. Thus, the pattern of the distribution of estimated CQI and real CQI are able to be observed in Fig. 14. It can be seen that the distribution for different PMI is various. That is, for a certain estimated CQI, the possibility of the corresponding CQI is various with PMI. In fact, the variation is large. Therefore, when designing the mechanism, we pay more attention to the difference of each PMI. The rest of the other CQI distribution for different PMI is shown in Fig. 16.

We also investigate the impact of the cos^2_θ and CQI.

In Fig. 15, the distributions for different CQI, which is calculated as $\mathbb{M}(|h|)$, are distinguished in terms of estimated CQI and cos^2_θ as well. Real CQI is denoted by $\mathbb{M}\left(\frac{\frac{P}{|S|}|h_k f_k|^2}{1+\frac{P}{|S|}\sum_{i\in S\setminus k}|h_k f_i|^2}\right)$. It is noticing in Fig. 15 that the variation of CQI is associated with to quantization error, cos^2_θ. These observations can be further exploited for the design of the model in the chapter 4. In short, the relationship between the estimated CQI and real CQI is more complicated than ever before.

doi:10.6342/NTU201900453

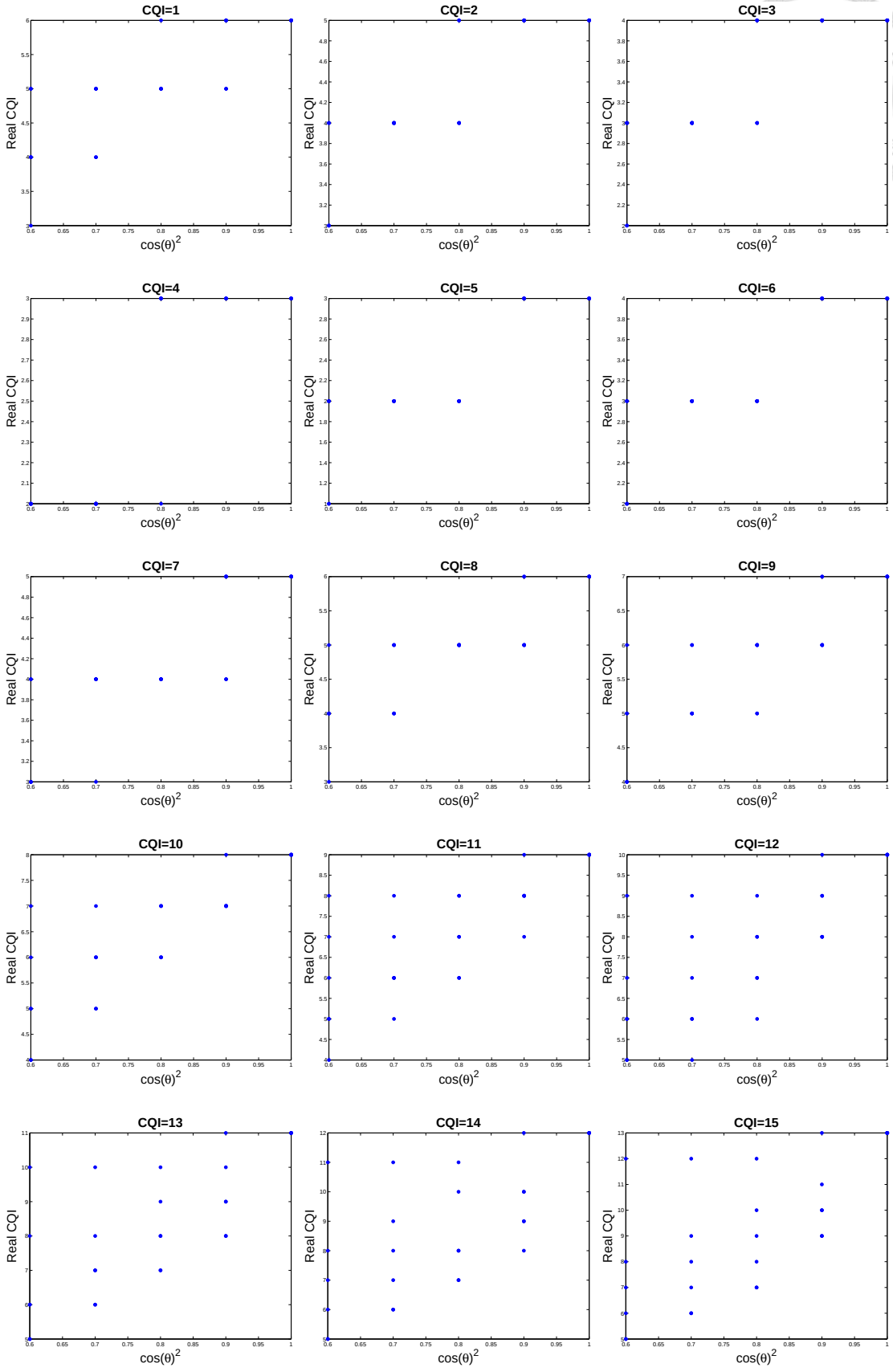


Figure 15: Real SINR in different Quantization Error(cos_θ) and Interference for different CQI. The dots in the same quantization error are represented as different interferences.

doi:10.6342/NTU201900453

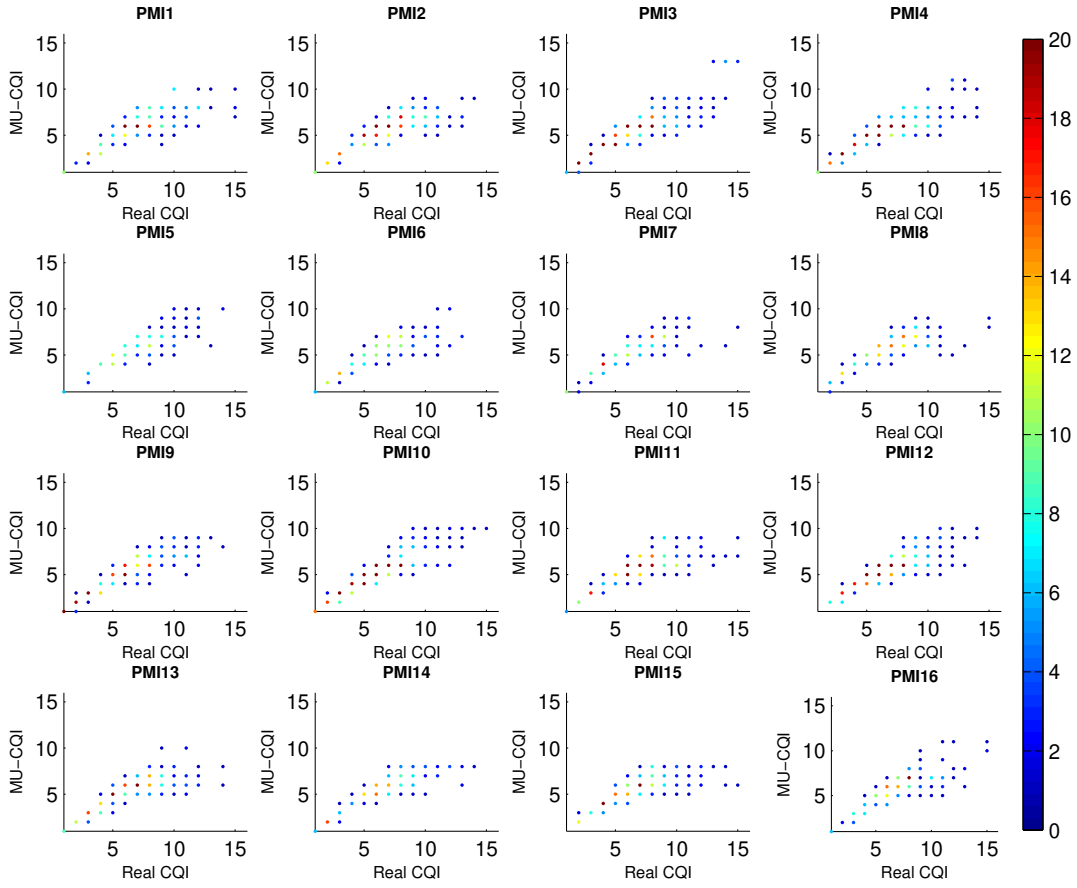


Figure 16: Real CQI and estimated CQI. Real CQI is calculated with channel vector h, the estimated CQI is the CQI return by UE. The estimated CQI is calculated following Eq. (3.12), which is a lower-bound in MU-MIMO cases.

doi:10.6342/NTU201900453

CHAPTER 4

PROPOSED REINFORCEMENT LEARNING BASED LINK ADAPTION

In this chapter, we would like to demonstrate how the reinforcement learning based link adaption work in the LTE communication system. Also, the implementation and design of the reinforcement learning technique will be further explained.

4.1 Motivation of Reinforcement Learning

We found that current studies regarding convergence are limited to the partial observations of problems. To be further explained, [12] said that dynamically change the step size can improve the performance. Larger step size can increase the convergence speed. It proposed a method to give a larger compensation step size while estimated BLER is large. [14] says that step size need to be large not only when the BLER is large, but also when BLER is far away for the target BLER region. And then, they will analyze the estimated BLER model or design certain decision maker based on their simulation environment. [30] said that the initial value is important, so it gathered large data in the beginning to get a proper initial value. However, gathering large data in the beginning is not always the possible in practice. It can be seen that all of these method are limited to their partial observations on the environment. And it looks like an endless work, there are always a new observation. However, reinforcement learning provide us the possibility to explore the strategy with less dependence on humans' intuition. Also, it is known for the capability of capturing the complicate relationship between the large parameters. In this case, we think that applying the RL technique properly can allow us to explore more strategy without humans' blind spot. What is proper initial value? When the step size have to be large? Except for the factors that previous research mentioned, is there any factors in the environment affecting the performance OLLA mechanism. What is the relationship between the initial value, step size, and information base station can get? Is there A well-designed reinforcement learning can explore the strategy in a more flexible and efficient way to improve the OLLA mechanism if the requirement is clear.

doi:10.6342/NTU201900453

Table 3: Notation Table for Reinforcement Learning

Symbols	Definition
s, s'	states
a	action
r	reward
$\mathcal{A}(s)$	set of all possible actions in state s
$\mathcal{A}$	set of all possible actions
$\mathcal{R}$	set of all possible rewards
s_t	state at time t
a_t	action at time t
r_t	reward at time t
G_t	return (cumulative discounted reward) following t
π	policy, decision-making rule
$\pi(s)$	action taken in state s
$\pi(a\|s)$	probability of taking action a in state s
$p(s', r\|s, a)$	probability of transition to state s' with reward r, from state s taking action a
$p(s'\|s, a)$	probability of transition to state s', from state s taking action a
$V_\pi(s)$	value of state s under policy π
$V_*(s)$	value of state s under optimal policy

doi:10.6342/NTU201900453

Taking a look into Eq. (3.13). We attempt to formulate all the relevant probability in Eq. (3.13). Thus, it can be rewritten into

$$\begin{aligned}
&\max_{\pi} \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=0}^{T} (r_{t+k} \mid s_t) \right] \\
&= \max_{\pi} \mathbb{E}_{\boldsymbol{\pi}} \left[\left(r_t + \sum_{k=1}^{T} \sum_{r_i \in \mathcal{R}} r_{t+k} \mid s_t \right) \right] \\
&= \max_{\pi} \sum_{MCS_i \in \mathcal{M}} P(MCS_i \mid s_t) \sum_{s_i' \in \mathcal{S}, r_i \in \mathcal{R}} P(s_i', r_i \mid s_t, MCS_i) r_i + \mathbb{E}_{\boldsymbol{\pi}} \left[\sum_{k=1}^{T} (r_{t+k} \mid s_t) \right].
\end{aligned} \tag{4.1}$$

Eq. (4.1) looks like a Markov decision process. However, the conventional methods, such as dynamic programming, are unable to solve the optimal function because the model is unknown to the agent. Thus, the transition probabilities are unknown. Nevertheless, if we take a look at the concept of reinforcement learning, we can see that the problems, which reinforcement learning aims to deal with, are similar to the problem in the thesis. Reinforcement learning is used to find a policy that achieves maximal reward over a long run. A bunch of researchers has focused on the decision-making issues with reinforcement learning. Thus, we can adopt suitable decision-making techniques among these researches. And then, we make some modifications to optimize the performance while adopting reinforcement learning.

Following the basic concept of reinforment learning [27], the framework of problem formulation, which is suitable for reinforcement learning, is defined as below,

$$\begin{aligned}
v_*(s) &= max_\pi \mathbb{E}_\pi \left[G_t | S_t = s \right] \\
&= max_\pi \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S_t = s \right] \\
&= max_\pi \mathbb{E}_\pi \left[R_{t+1} + \sum_{k=0}^{\infty} \gamma^k R_{t+k+2} \mid S_t = s \right] \\
&= max_\pi \sum_a \pi(a|s) \sum_{s',r} p(s', r|s, a)[r + \gamma v_\pi(s')],
\end{aligned} \tag{4.2}$$

where π is denoted by the policy of selecting the action, a, based on the state, s, which is denoted by the observation from the environment. $\pi(a|s)$ is the probability of the policy selecting a in s. G_t is the cumulative return function, which could be defined according to the situation. Basically, if any problems can be transformed into this form, it is proper to adopt reinforcement learning methods to solve the problem.

doi:10.6342/NTU201900453

In comparison with Eq. (4.1), these two formulas looks similar. We can easily transform Eq. (4.1) into the form of problem in the reinforcement learning, Eq. (4.2). G_t, $\pi(a \mid s)$, $p(s', r|s, a)$ and can be expressed as,

$$\begin{aligned} G_t &= \sum_{k=0}^{T} r_{t+k} \\ \pi(a \mid s) &= P(MCS_i \mid s_t) \\ p(s', r|s, a) &= P(s', r_i \mid s_t, MCS_i) \end{aligned} \tag{4.3}$$

From the above equations, we can see that the problem formulation is able to be easily transformed into the form of a reinforcement learning problem.

In our case, s is the information about the environment, which is observable to the base station, as well as the historical information. The design of s will be further explained in the next chapter. a is the actions the base station can make. In this case, $a \in \mathcal{M}$.

4.2 Train an OLLA Agent based on Reinforcement Learning

In this section, we will elaborate how do we train a agent to to assigned MCS based on the observation of the environment, realizing the OLLA mechanism.

In this work, tensor flow [31] and keras [32], which are developed based on python, are adopted to train the agent. As a result, we have to duplicate the crucial part of the communication environment in VIENNA for the Python platform in order to train the agent. Although our environments are simulated through VIENNA [21], there are advantages of duplicating the communication environment in python instead of applying the machine learning algorithm in VIENNA. One is that it can save our plenty of time to implement the machine learning algorithm, which is a really complicate and delicate work. Our major work is to train a agent, which is suitable for the problem in the this, rather than implement the machine learning algorithm. For another reason, training agent in VIENNA could cost much more time, because a communication simulator has lots of works, such as scheduling, calculating channel response, waiting for the response of UE or the base station, and so on. With carefully duplicating the essential part of the environment into the Python platform could save us plenty of time. The details of how we duplicate the environment in VIENNA will be elaborated in the following.

We have known that the success of the transmission can be decided by the mismatch gap between real SINR and the tolerable SINR of certain MCS. If the mismatch is large, it means that the assigned is either too aggressive or conservative. That is to say, once the real channel is known, we can predict if the feedback

doi:10.6342/NTU201900453

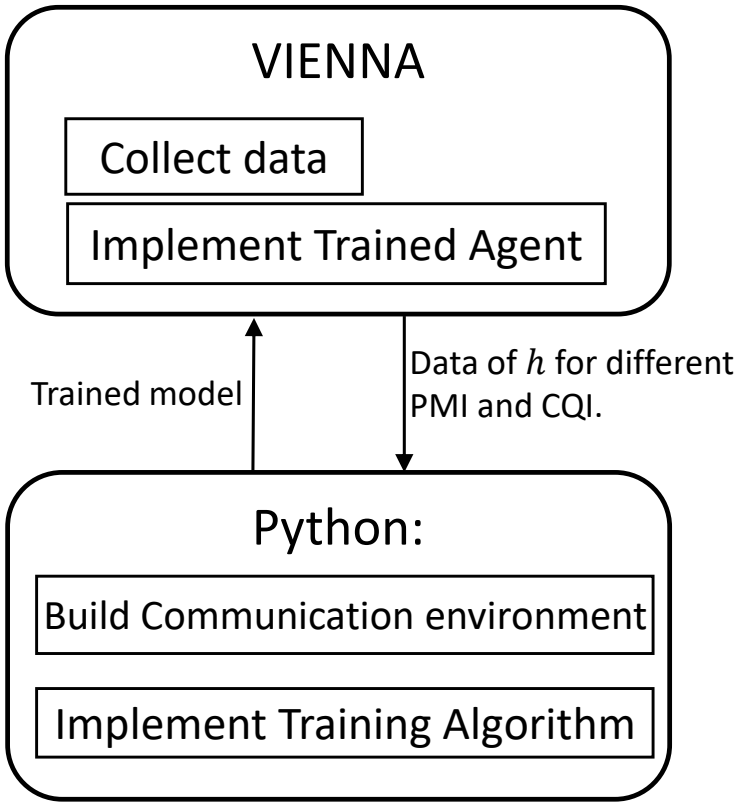


Figure 17: Training Procedure for the proposed Algorithm

is NACK or ACK without waiting for response UE, which is a time-consuming process. As a result, we can gather all the necessary data from the simulator before training. The simulator used in the thesis is VIENNA. The simulator is able to generate the channel response based on the setting. Although the base station is unable to know the perfect channel information, we still can find the information of the perfect channel information in the simulator. Thus, channel responses can be stored and utilized afterward. Fig. 17 demonstrate the components we implemented in VIENNA and Python, respectively. The training process is listed in the following:

1. Simulator generates all the necessary data, including the real channel response h_k, PMI, and CQI (the feedback estimated by the UE), for training.
2. Duplicate the essential part of the communication system, built the environment in python.
3. Train the agent in python based on reinforcement learning algorithm.
4. Implement the trained agent as mentioned in Fig. 18.

doi:10.6342/NTU201900453

While exploiting the machine learning technique, the data have to be carefully obtained. Otherwise, the trained agent could be biased due to the improper database. This criterion also applies to reinforcement learning because it is also one of the machine learning technique. Thus, we run the simulation with the random assignment of PMI. The assignment of SNR for each UE is based on the distribution of SNR given by MTK.

It is noticing that when training the agent, the agent will do exploration and exploitation and update is policy through interactions. By contrast, trained agent will on exploit current policy and do not update the policy. The relationship between the trained agent and the other components in communication system is shown in the next section.

4.3 Proposed Mechanism in Communication System

Fig. 18 shows how we implement the trained agent in the communication system. Firstly, the UEs return their channel information(CSI) following the standard of LTE including PMI and CQI. The diagram block, which says 'Modify the estimated CQI with Trained Agent', is where the proposed method activates. In this diagram, it considers not only CQI but also the historical data and HARQ information returned by UE. And then, it will produce the modified CQI'. The scheduler applies the CQI' to estimate the channel capacity of the UEs. Then, the scheduler uses the information to calculate the efficiency of the different combination, and choose the best combination based on following its policy.

Our mission is to train a agent, which is capable to choose MCS in response to their knowledge of the environment. Thus, we focus on the diagram block, which says 'Modify the estimated CQI with Trained Agent'. In the case, the information of the base station is the feedback of CSI, ACK/NACK and the historical data preserved in the base station. Every time the base stations assign a selected MCS to the scheduled UE, the base stations record that the scheduled UE. And then, the scheduled UEs return ACK or NACK for the MCS assigned to them.

The next section demonstrated how we train the agent.

4.3.1 The Design of State, Reward, and Neural Network

4.3.1.1 State and Reward

The basic components in reinforcement learning are s(state), r(rewarding), and a(action) as we have introduced in the chapter 2. s includes the observable parameters for the agent only. a is the decision made by the agent according to the observable s. That is, $a = \pi(s)$. In the communication system, s is the

doi:10.6342/NTU201900453

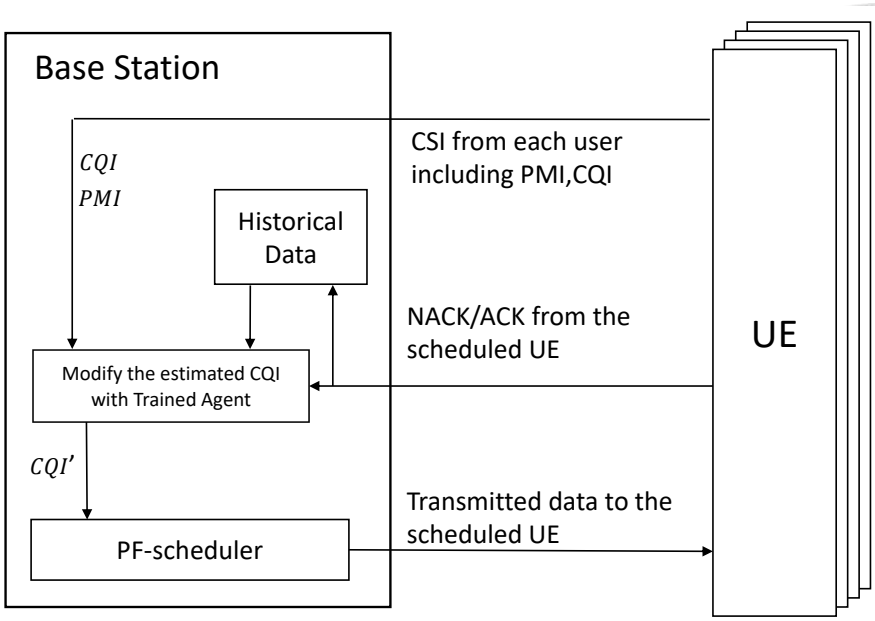


Figure 18: Diagram of proposed Mechanism in Communication System

information that the base station can obtain. We define a as the MCS selected in next TTI according to s.

The designing process of s is very similar to the feature extraction since the output a is highly dependent on the s. It is known that the feature extraction has a significant impact on the performance of the system. Thus, we will discuss how we design s. The features have to include the relevant factors and exclude irrelevant factors as much as possible. In addition, the normalization of the feature is also an important issue. In practical, it is hard to know which feature is irrelevant or relevant. Nevertheless, having the knowledge of the problem is very helpful for overall performance. In this work, our target function is defined as Eq. (4.1). That is, we want to find the target MCS as fast as possible. Intuitively, the historical data, which indicating the record of the assigned MCSs and the response (ACK/NACK) of the assigned MCSs, have an impact on the next selection of MCS. Thus, s should take the historical data into consideration. In this work, the form of historical data is designed carefully. To the best of our knowledge, we have known that the next selected MCS should be larger the maximal known MCS, which can be transmitted successfully. Likewise, the next selected MCS should be smaller than the minimal MCS, which leads to failed transmission. As a result, instead of recording all transmission results, we attempt to simplify the form of recording historical data. Without simplification, the features are

$$\begin{bmatrix} ACK & NACK & \text{Not used} \end{bmatrix}^{N_{comb} \times N_{MCS}}.$$

The base station maintains the matrix for each user to record the result for

doi:10.6342/NTU201900453

each MCS while co-scheduling with different users. The result could be ACK or $NACK$. For the MCSs have not been assigned can be recorded as Not used.

With simplification, the matrix for each user becomes

$$\begin{bmatrix} maxAckMCS & minNackMCS \end{bmatrix}^{N_{comb}},$$

which records only the maximal MCS receiving ACK and minimal MCS receiving NACK within assigned MCSs. In theory, eliminating irrelevant feature should improve the performance and prevent it from overfitting. The simplified matrix is chosen in the thesis because it shows better capability. The results are verified in chapter 5 in Fig. 32.

Based on our analysis, the combination of the other beams has an influence on the desired MCS. Furthermore, it can be observed in Fig. 15 that if the $\cos\theta$ is known, the range of possible MCSs will be narrowed down. For example, assuming there are five possible co-scheduled PMI for a certain UE. If we can get suitable MCS corresponding to the UE scheduled with different PMI. The more accurate MCSs can we get, the narrower the range of $\cos\theta$ is. As a result, the possible range of MCS for the rest unknown combination can be smaller. That is to say, knowing the correct MCSs of UE scheduled with distinct PMIs might accelerate the speed of finding MCS.

Also, since it can be observed in Fig. 16 that different PMI has distinct characteristics, we consider PMI as one of the features.

In short, we put the historical data, the PMI and CQI of the user, and the PMI of the co-scheduled user into s. Also, how to transform the feature is a crucial issue. PMI and CQI are treated as category features, we transform them by hot-encoder, which is a well-known encoder for category features. After several comparison and experiments, our final choice of s is shown as Eq. (4.4). The comparison of different design of features is shown in chapter 5 in Fig. 32.

Fig. 19 demonstrates how the proposed OLLA actually works according to the proposed feature.

The rewarding function indicates the score from s_i to s_j while applying a. Since our aims are to train a agent able to be applied to the base stations, the rewarding rule has to take the practical environment into consideration.

The base station is unable to make sure about if the assigned ACK is correct or not until it finds the lower bound of unavailable MCS and upper bound of the available MCS. Thus, we take *else if* $|maxAckMCS_{i,t}^{Pair_{i,j}} - minNackMCS_{i,t}^{Pair_{i,j}}| \leqslant 1$ as a stop criteria. Also, the last assigned MCS should be available so the last assigned MCS should be available. It can be seen that the punishment score is different according to the condition. Since it is obvious that the MCS, which is higher than current unavailable MCS and lower than current available MCS,

doi:10.6342/NTU201900453

Table 4: Design of s, r, and a

State:

$$\mathrm{s}_{i,t} = \left[\left[maxAckMCS \quad minNackMCS \right]_{i,t}^{N_{comb}} \quad Pair_{i,j} \quad CQI_{i,t} \quad PMI_{i,t} \right]. \tag{4.4}$$

Reward:

$$\mathrm{r}_{i,t} = \begin{cases} -6, & \text{if } a_{i,t} < maxAckMCS_{i,t}^{Pair_{i,j}} \text{ or } a_{i,t} > minNackMCS_{i,t}^{Pair_{i,j}} \\ 0, & \text{else if } |maxAckMCS_{i,t}^{Pair_{i,j}} - minNackMCS_{i,t}^{Pair_{i,j}}| \leqslant 1 \\ & \text{and } a_{i,t} == maxAckMCS_{i,t}^{Pair_{i,j}} \\ -1, & otherwise \end{cases} \tag{4.5}$$

Action:

$a_{i,t}$ is the selected MCS in next TTI according to $s_{i,t}$ for UE_i. $a_{i,t} \in \mathcal{M}$.

Parameters	Description
$maxAckMCS$	The maximal MCS is able to get ACK within MCSs, which have been allocated previously.
$minNackMCS$	The minimal MCS is able to get NACK within MCSs, which have been allocated previously.
N_{comb}	The maximal possible number of co-scheduled PMI.
N_{MCS}	The number of the MCSs
$\mathcal{M}$	Set of MCSs
$CQI_{i,t}$	The CQI of the scheduled UE_i.
$PMI_{i,t}$	The PMI of the scheduled UE_i.
$PMI_{j,t}$	The PMI of the co-scheduled UE_j.
$Pair_{i,j}$	The index of the Pair while the scheduled UE_i is co-scheduled with UE_j.
$a_{i,t}$	The selected MCS for UE_i .
$realSINR_{i,j}$	The real SINR of UE_i while it is co-scheduled with UE_j.

doi:10.6342/NTU201900453

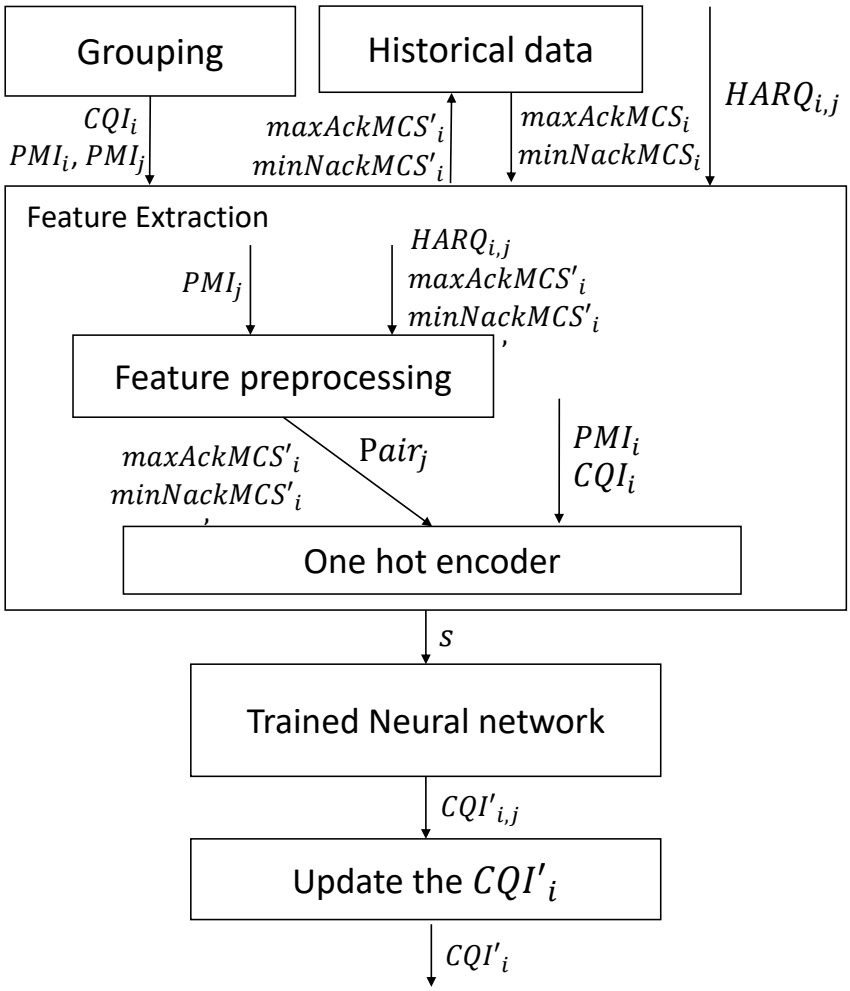


Figure 19: Block of 'Modify the estimated CQI with Trained Agent' in Fig. 18 in Detail

should never been tried, the punishment is given more. The rewarding function can be represented as

$$r_{i,t} = \begin{cases} -R, & \text{if } a_{i,t} < maxAckMCS_{i,t}^{Pair_{i,j}} \text{ or } a_{i,t} > minNackMCS_{i,t}^{Pair_{i,j}} \\ 0, & \text{else if } |maxAckMCS_{i,t}^{Pair_{i,j}} - minNackMCS_{i,t}^{Pair_{i,j}}| \leqslant 1 \\ & \text{and } a_{i,t} == maxAckMCS_{i,t}^{Pair_{i,j}} \\ -1, & otherwise \end{cases} . \tag{4.6}$$

The improper magnitude the rewarding might lead to failed converge of the agent. We find the proper magnitude of the rewarding throughput experiments.

doi:10.6342/NTU201900453

The experiment result is shown in Fig. 37 and Fig. 38 . Furthermore, we investigate the effectiveness of the rule, $-R$, if $a < maxAckMCS_{i,t}^{pair_{i,j}}$ or $a > minNackMCS_{i,t}^{pair_{i,j}}$. It seems that it has a positive impact on the performance.

In addition, although our major goal is to minimize the converge steps, it might turn out that the strategy of minimizing converge step will lead to higher BLER in the condition that the optimal strategy tends to select the MCS higher than the maximal available MCS. To avoid this condition, we can modify the rewarding function to control the BLER. The transmissions fail under the condition that the that the assigned MCS is higher than the real SINR. Thus, we can control the BLER by changing the punishment of the condition the MCS is higher than the real SINR. The overall design of rewarding can be represented as

$$r_{i,t} = \begin{cases} -6, & \text{if } a_{i,t} < maxAckMCS_{i,t}^{Pair_{i,j}} \text{ or } a_{i,t} > minNackMCS_{i,t}^{Pair_{i,j}} \\ 0, & \text{else if } |maxAckMCS_{i,t}^{Pair_{i,j}} - minNackMCS_{i,t}^{Pair_{i,j}}| \leqslant 1 \\ & \text{and } a_{i,t} == maxAckMCS_{i,t}^{Pair_{i,j}} \\ -1, & \text{else if } a_{i,t} \leqslant realSINR_{i,j} \\ -R_{nack}, & \text{else if } a_{i,t} > realSINR_{i,j} \end{cases} . \tag{4.7}$$

The effectiveness of the modification is presented in Fig. 52. The overall design of the s, r ,and a are shown in Table. 4.

4.3.1.2 Neural Network

In this work, we apply a neural network as the approximate function.

In comparison with previous OLLA methods, although applying neural network shows better performance, it requires more arithmetic units. Hence, several researches aimed to improve the energy and computation speed through the design of the hardware [33–36]. Parallel computing [33] can improve the computing speed considerably. Due to the limitation of the communication devices, [35] provided an overview of trends in designing machine learning architecture Thanks for these efforts and the trend, we can expect that the cost of applying neural network in communication will be reduced.

Secondly, neural network has stronger flexibility of representing complicate relationship between parameters. Training it properly can reduce human resources costs. For example, Fig. 20 shows that no matter what kind of the feedbacks the UE adopted, the proposed method can find the strategy automatically.

Furthermore, due to the characteristics of representing complicate relationship between parameters, it provides the potential of multiple to multiple map as shown in Fig. 21. With the increasing complexity of communication environment, such

doi:10.6342/NTU201900453

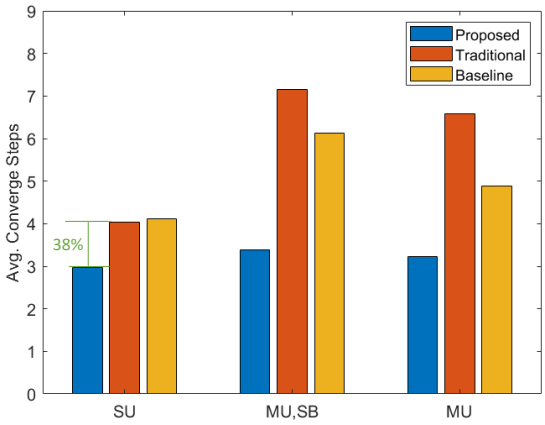


Figure 20: Comparison of Convergence Steps between different Methods in different Types of Feedbacks

as multiple antennas and increasing devices, the requirement of making good use of the increasing parameters for base station is growing.

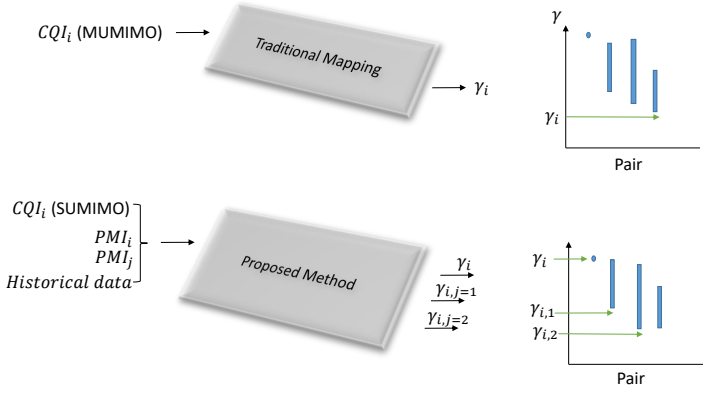


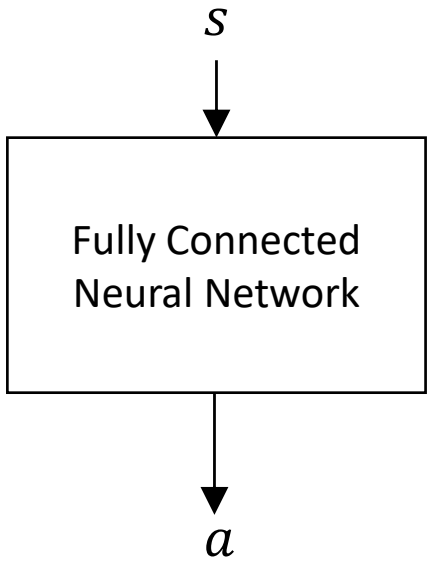
Figure 21: Difference between traditional Mapping and proposed Method with good initial Value

When it comes to applying neural network, properly determining the parameters and architecture is crucial for good training. Fortunately, there have exited several useful thumbs of rules about the setting. Although some of the issues are still under discussion, this discussion shows a path to train a agent in an efficient way.

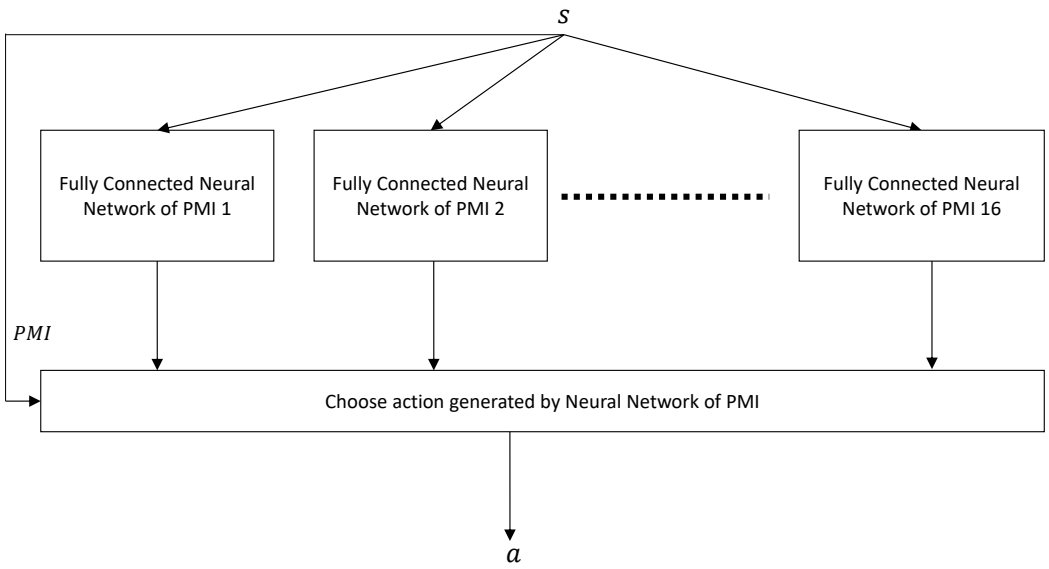
In this thesis, we try two types of the architecture of the neural network as shown in Fig. 22. The PMIs have distinct characteristics, so we separate the neural network as shown in Fig. 22. The the architecture of fully-connected network is the same as Fig. 23 but without PMI_i. We hypothesis this architecture should have a stronger capability of capturing the characteristics PMI more carefully and

doi:10.6342/NTU201900453

efficiently. The result is shown in Fig. 34.



(a) One fully-connected Network.



(b) Multiple fully-connected Network for each PMI.

Figure 22: Architecture of the Neural Network

The fully-connected network in this thesis is demonstrated in Fig. 23. $_i$ is the weight between layer (i-1) and layer i. X_i represents the set of the output of neurons in layer i. X_{ij} represents the output of neuron j in layer i. Z_i represents the set of input of neurons in layer i. b_i is the bias in the layer i.

$h_i(Z_i)$ is activate function. The relationship between X_i and Z_i is $X_i = h_i(Z_i)$.

doi:10.6342/NTU201900453

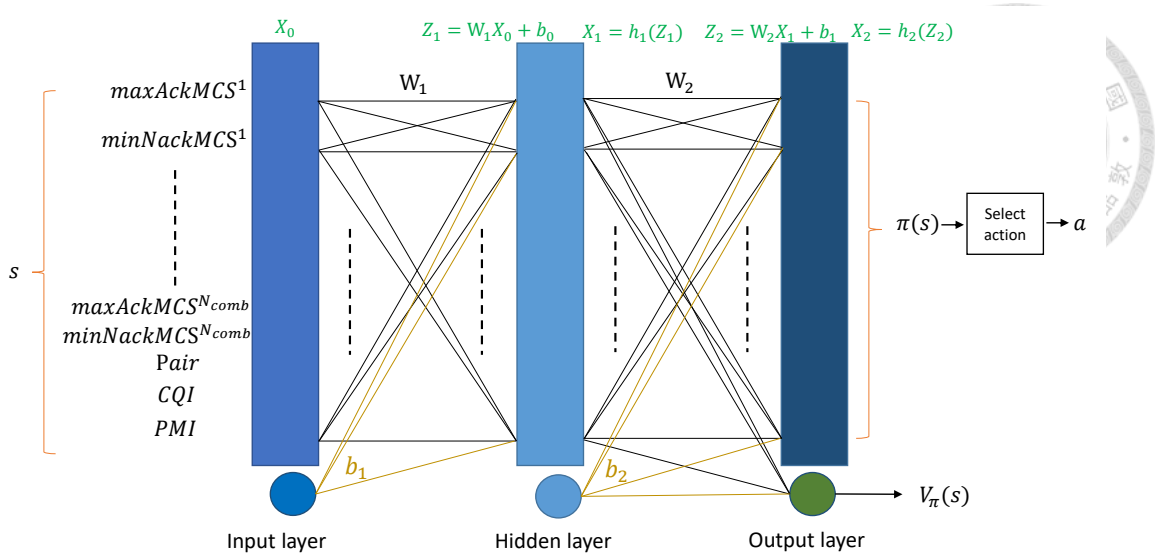


Figure 23: Fully-connected Network

We adopt RELU as the activate function in the hidden layer. RELU function is

$$RELU(Z) = max(0, Z).$$

The RELU is popular in recent years. It can accelerate the convergence of stochastic gradient descent compared to the sigmoid/tanh functions. Furthermore, while considering multiple layers, vanishing gradient problem will lead to the failure of deep learning. It is shown that substituting the activation function with RELU can solve this issue. Due to these benefits, we choose RELU.

In the output layers, we adopt softmax layer for $\pi(s)$, which outputs the probability of choosing actions. The neuron j in output layer can be written as

$$softmax(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}},$$

where K is the number of the actions. In this work, $K = |\mathcal{M}|$.

The sum of the value of the neurons in the layer is equal to 1, the values are always between $[0, 1]$. Thus, each output can be treated as probability easily. It is common for action selection based on the stochastic policy. As for $v_{pi}(s)$, the activate function is linear function, which is

$$linear(Z) = Z.$$

Let input layer as layer 0. X_0 is the input directly. Also, We treat all the input as category feature, so we encoded the features with one hot encoder. Take PMI for example, if $PMI = 16$, after encoded, it becomes $[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$. That is, if $PMI = k$, only the kth value in the array will be 1, the others will be

doi:10.6342/NTU201900453

zeros. Since the category feature has no numerical meaning, thus proper transform is needed. Although some machine learning technique can make a good prediction with numerical form, the neural network is sensitive to the numerical value. Thus, it may be better to apply the one hot encoder in this case.

The effectiveness of the feature can be seen in Fig. 32.

4.4 Reinforcement Learning Algorithm

There are many categories of reinforcement learning. It can be simply separated into a value-based method, policy-based method, and actor-critic method. The detailed explanation can be found in [27]. After our investigation, we focus on exploring methods adopt actor-critic because it has the advantage of both value-based and policy-based methods. Apart from the learning algorithm, the approximate function is also an issue. We choose the neural network to learn the approximate function. We firstly introduce the actor-critic method applied in the thesis. And then, the details of the neural network and the exploration-and-exploitation policy are elaborated.

4.4.1 Asynchronous Advantage Actor-Critic Agents(A3C)

We adopt A3C [37] as the training algorithm. A3C recently receives considerable attention. This algorithm has better performance in comparison with the current state of art reinforcement learning algorithms in many aspects. In order to have a basic understanding of the algorithm, we will explain why it is called A3C.

Asynchronous indicates that the whole updates of the network do not operate immediately as long as the reward is received. Instead, in A3C, there are many agents with corresponding environment operating in parallel. In this way, the algorithms are able to explore more strategies since each agent is independent with of the experience of others. It keeps a good trade-off between exploration and exploitation, which is an important issue for reinforcement learning.

Actor-Critic is a combination of the policy-based and value-based method. The method utilizes the benefits of traditional value-based and policy-based method, improving their weakness. Thus, Actor-Critic uses the parameterized value function, v_π, and parameterized policy function, π, to learn the approximate function in a more efficient way. In spite of the benefits of combining value-based and policy-based method, updating v_π and π is not feasible in practice; because they can affect each other, that is why the new algorithms, such as A3C and A2C, are appeared.

In a conventional policy-based method, it is general to use a rewarding function,

doi:10.6342/NTU201900453

which simply is an indicator of how good the action is, for updating gradients. However, A3C adopt **Advantage** function as an indication. **Advantage** function indicates how much better is this update rather than how good the actions are. It is generously written as, $A = R - v_\pi(s)$. Thus, it makes the algorithm more robust to the biased issue and tunes the network in a more efficient way as well. Nevertheless, it enlarges the impact of the variance of the reward on learning. Thus, the design of the reward still remains an important issue.

4.4.2 Implementation and Modification of A3C

In this section, the details of the implementation of A3C will be further explained. The overall algorithm is shown in Algorithm 1.

Algorithm 1 N-step A3C with shared neural network and ϵ-greedy policy in the thesis.

\\ Assume global shared parameter vectors θ global shared counter $T = 0$ \\ Assume thread-specific parameter vector θ' \\ c_v and c_{reg} are constants. Initialize thread step counter $t \leftarrow 1$ Reset loss: $L \leftarrow 0$ Synchronize thread-specific parameters: $\theta' = \theta$ $t_{start} = t$ Get state s_t Perform a_t according to ϵ-greedy algorithm, as depicted in Algorithm 2. Receive reward r_t, which is calculated based on Eq. (4.5), and new state s_{t+1} $t \leftarrow t + 1$ $T \leftarrow T + 1$ terminal s_t $t - t_{start} == N$ $R = \begin{cases} 0, & \text{for terminal } s_t \\ V(s_t, \theta'), & \text{for non-terminal } s_t \end{cases}$ $i \in t-1, ..., t_{start}$ $R \leftarrow r_i + \gamma R$ Advantage function : $A = R - V(s_i, \theta')$ $L_{v,i} = (A_i)^2$ $L_{\pi,i} = -log(\pi(a_i|s_i)A_i$ $H(\pi(s_i)) = -\sum_{k=1}^{|\mathcal{A}|} \pi(s_i)_k \cdot log\pi(s_i)_k$ $L_{reg,i} = H(\pi(s_i))$ $L_i = L_{\pi,i} + c_v L_{v,i} + c_{reg} L_{reg,i}$ Accumulate loss : $L \leftarrow L + L_i$ Calculate gradient : $d\theta \leftarrow \bigtriangledown_{\theta'} L$ Perform asynchronous update of θ using $d\theta$ $T > T_{max}$

Although the A3C offers a robust structure for reinforcement learning, there are still some important components has to be further discussed as applying it in a new field.

Firstly, the approximation functions, $v_\pi(s)$ and $\pi(s)$, are the important components. The approximation function has to be parameterized and differentiable since A3C apply policy gradient theorem. We adopt the neural network as approximation functions $v_\pi(s)$ and $\pi(s)$. Neural networks are good at extracting the feature from complicated inputs. Also, the gradient theorem for the neural network has been well studied. Nevertheless, the activation layers and architecture of neural networks still remain issues because they highly depend on the type of tasks. The neural network is picked in this thesis as approximation functions, $v_\pi(s)$ and $\pi(s)$. The performance of different structure will be demonstrated in the next chapter.

It is noticing that $v_\pi(s)$ and $\pi(s)$ share same neural network in our work, as

doi:10.6342/NTU201900453

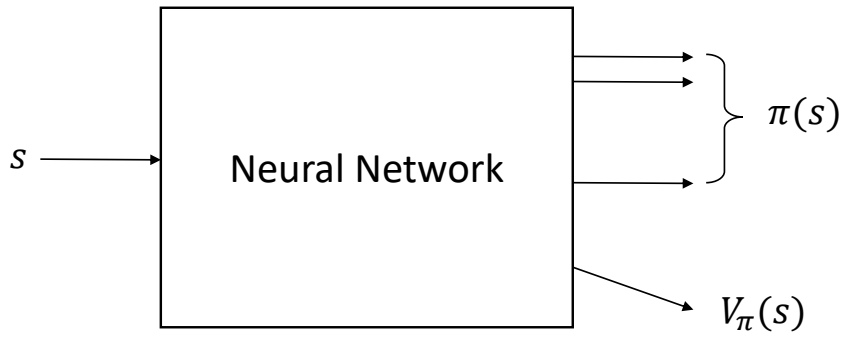


Figure 24: Architecture of Neural Network of s, $\pi(s)$ and $v_\pi(s)$. The block 'Neural Network' in this thesis is the same as Fig. 22.

shown in Fig. 24. This design provides some benefits. Optimizing the neural network together act as a regularization, and result in a more stable network. Plus, optimizing both goals together will learn faster [37].

Secondly, the task of taking action is the exploration and exploitation issue. Exploiting the current network or exploration the other possible policy is always a question. Instead of only relying on softmax action selection, we adopt ϵ-greedy to keep a balance in this problem. In the exploitation stage, the agent still acts based on softmax action selection. The overall procedure for deciding to explore or exploit is shown in Algorithm.2.

Algorithm 2 Peseudo code of Selecting Actions in Agents

\\ ϵ_0 is the initial value of ϵ; ϵ_{end} is the end value of ϵ; ϵ_t is the value of ϵ at step t \\ t is current number of training step; ϵ_0 will reach ϵ_{end} after T steps
$t < T$ $\epsilon_{t+1} = \epsilon_t - (\epsilon_0 - \epsilon_{end})/T$ $\epsilon_{t+1} = \epsilon_{end}$ i =random number picked from [0,1]
$i < \epsilon_{t+1}$ a_{t+1} picked randomly from [($maxAckMCS$+1),($minNackMCS$-1)]
a_{t+1} selected according to the current policy.

It can be seen in the Algorithm 2 that it is a ϵ-greedy. Although it is general to use stochastic policy based on the softmax layer for exploration and exploitation, the action space can not be easily constrained and well explored as shown in Fig. 36; thus, we adopt ϵ-greedy. In addition, even if it is an exploration stage, we still give constraints for the action space. We found that in this way, the exploration will be more efficient. Since the action spaces are extremely large, providing the reasonable selection space of MCS based on the known knowledge can encourage the optimizer to train the neural network along with a desirable direction.

doi:10.6342/NTU201900453

Otherwise, without the constraints, it may receive a bunch of discouragement, the direction of gradients could be hard to predict. As a result, the convergence speed will be slow down. Still, the neural network has to learn what kind of actions is discouraged. Thus, we put no constraints on the exploiting policy. Although the neural network may return an undesirable action, these undesirable actions may teach the training agent in which actions are bad for the state. Besides, if the discouraged actions are never outputted by the neural network, the agent is unnecessary to worry about it. In short, this design explores the policy more efficiently and allows the updating algorithm to specifically deal with the weakness of the neural network The comparison is shown in Fig. 36.

Moreover, the loss function is a crucial element in neural network. The neural network applies backpropagation to update the parameters in the network based on the loss function. The design of the loss function has impact on the convergence speed, exploration, and the way the neural network converging. The loss function can be separated into three parts: policy loss(L_π), value loss(L_v) and regularization(L_{reg}). The loss function L is

$$L_i = L_{\pi,i} + c_v L_{v,i} + c_{reg} L_{reg,i}, \tag{4.8}$$

where c_v and c_{reg} are coefficients. It can be modified to tune the impact of L_v and L_{reg}.

L_π depicts the loss of the policy, thus, it can be written as

$$L_{\pi,i} = -log(\pi(a_i|s_i)A_i, \tag{4.9}$$

where A_i is the advantage function. $A_i = r_i + \gamma r_{i+1} + ... + \gamma^{N-1} r_{i+N-1} + \gamma^N V_\pi(s_{i+N}) - V_\pi(s_i)$. n is number of samples.

Value loss function depict how accurate is the prediction of the value function. It can be simply represented as the advantage function:

$$L_{v,i} = (A_i)^2. \tag{4.10}$$

It is found that adding the entropy of policyπ can improve the objective function because it discourages π to choose the premature action. As a result, the exploration is improved.

In this work, we adopt cross entropy as entropy function. Entropy function is defined as

$$H(\pi(s_i)) = -\sum_{k=1}^{|\mathcal{A}|} \pi(s_i)_k \cdot log\pi(s_i)_k,$$

where $\pi(s)_k$ is the probability of choosing action k.

doi:10.6342/NTU201900453

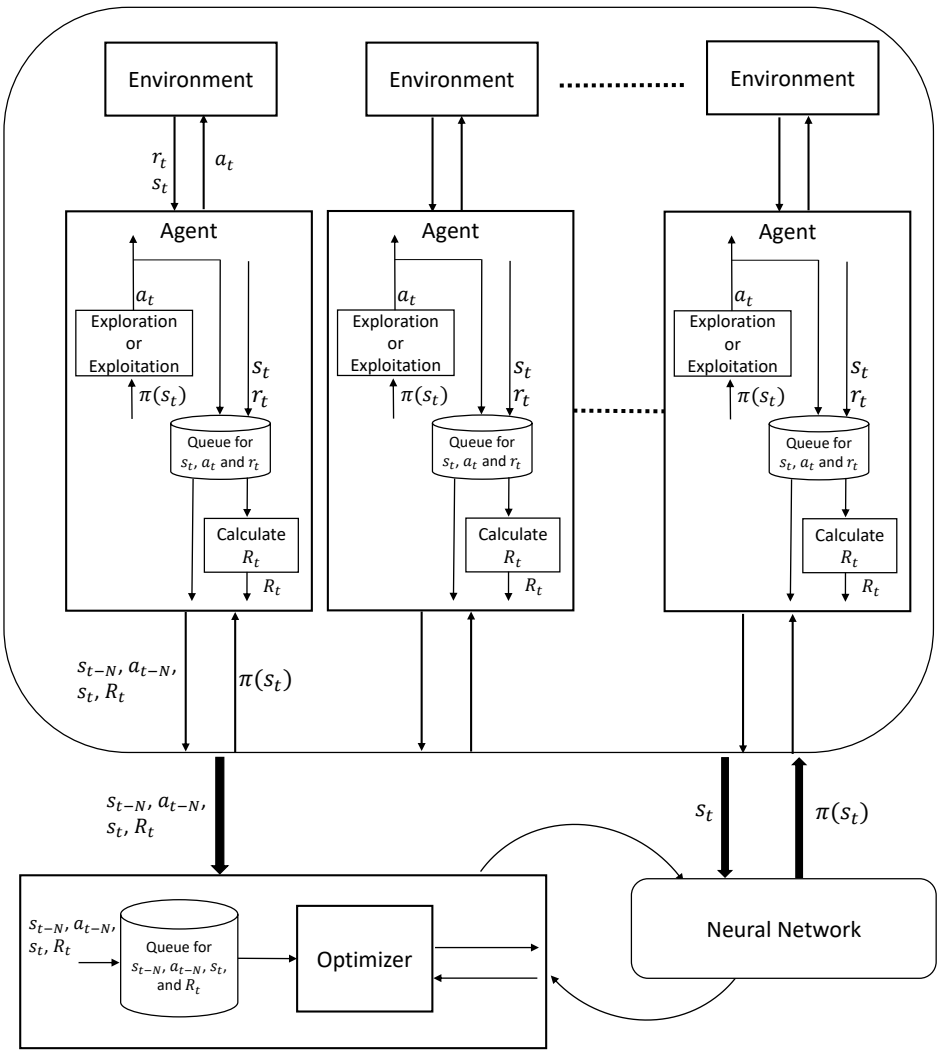


Figure 25: Implementation of A3C in training Neural Network

Also we can use this term,

$$L_{reg,i} = H(\pi(s_i)), \tag{4.11}$$

for regularization.

The neural network will update the parameters in the network based on the L_i we have discussed.

We also investigate the impact the parameter, N, mentioned in Algorithm 1 - N-step A3c algorithm. We compare the performance in terms of the objective function, Eq. (3.13). N-step return is the combination of temperal difference(TD) and Monte Carlo(MC). With n-step return, the V_π is updated according to this formulation, $V_\pi(s_t) = r_t + \gamma r_{t+1} + ... + \gamma^{N-1} r_{t+N-1} + \gamma^N V_\pi(s_{t+N})$.

Reward function in Fig. 25 is defined as $R_t = r_t + \gamma r_{t+1} + ... + \gamma^{N-1} r_{t+N-1}$, so the advantage function is $A_t = r_t + \gamma r_{t+1} + ... + \gamma^{N-1} r_{t+N-1} + \gamma^N V_\pi(s_{t+N}) - V_\pi(s_t)$, where $\gamma^N V_\pi(s_{t+N})$ is for bootstrapping, and $V_\pi(s_t)$ is served as baseline.

doi:10.6342/NTU201900453

Research has shown that while considering a sequence of actions, N-step returns may accelerate the learning speed. In n-step return, the function can be updated with cumulative n-step rewards in each iteration, benefitting from the unexpected magnitude of rewards. The 1-step return has an only 1-step reward, so it changes the functions much slower. Nevertheless, it is not always the case that n-step returns will accelerate the convergence speed. The variance of n-step rewards highly depends on the chain of actions, which may lead to lots of different combinations of states. If the size of the possible state is too large, it might endanger the convergence. Thus, the proper chosen of N have to take into consideration [27]. The comparison of N is shown in Fig. 37.

The details of the functions in the algorithm have been discussed. Now, each block in Fig. 25 will be introduced.

The task of the environment is to emulate the behavior of the communication network. Moreover, it has to generate appropriate reward and state in response to the action of the agent. For the reason that we focus on solving the strategy of assigning MCS, the environment is used to generate NACK and ACK for the assigned MCS. It estimates the real SINR with the collected database and decides whether the selected MCS would lead to NACK or ACK by comparing the real SINR and the tolerable SINR for the MCS according to map.

The task of the agent is to make an action. Generally, each agent calculates their own local gradients, and transmit these gradients to global optimizer after N steps. And then, the global optimizer will update all the parameters in the network based on the gradients from each agent in A3C.

In this thesis, we made a little adjustment without losing the benefits of the A3C algorithm. Except for making an action, the agents store all states, rewards, and taken actions. And then, the agents transmit s_0, $s_N||s_{terminal}$, and R for each N steps to global optimizer. The difference is that the agents no longer compute the gradients on their own. Nevertheless, the global optimizer is still able to receive the experience from different agents, so the explorations remain effective. Besides, due to the optimizer happening only in the global optimizer, it can save the time of computing gradient, which is a heavy burden for computers. The details of updating neural network is shown in Fig 26.

4.5 Proposed Feedback and Scheduler

In the framework of OLLA, we have two suggestions. Firstly, we suggest that using the feedback considering only a single beam instead of multiple beams in order to improve the overall performance. This type of feedback is called SU-MIMO feedback.

doi:10.6342/NTU201900453

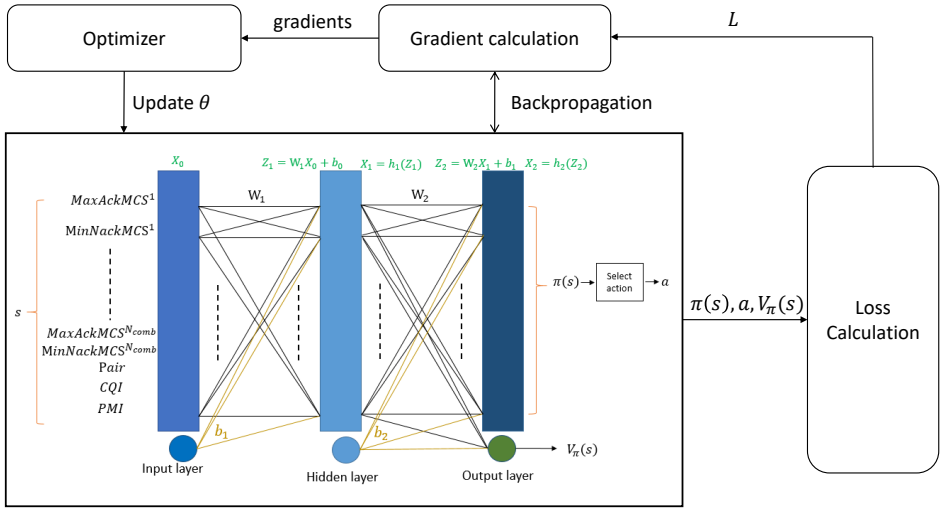


Figure 26: Process of updating a Neural Network. The gradient descent optimization algorithms in 'Optimizer' in this thesis is RMSprop. The backpropagation needs to compute the derivative of each activate function and the error generated in each layers.

To deal with the feedback problem in MU-MIMO, VIENNA adopts the feedback method in [3]. The approximation SINR in [3] is expressed as

$$\begin{aligned} E[SINR_{k,real}] =& \frac{\frac{P}{|S|}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k\right|^2}{1+\frac{P}{|S|}\left\|h_k\right\|^2 sin^2\theta_k\sum_{i\in S\setminus k}\left|e_k\widetilde{f_i}\right|^2} \\ \geq& \frac{\frac{P}{|S|}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k\right|^2}{1+\frac{P}{|S|}\frac{|S|-1}{M-1}\left\|h_k\right\|^2 sin^2\theta_k} \\ \geq& \frac{M}{|S|\left\|f_k\right\|^2}\frac{\frac{P}{M}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k\right|^2}{1+\frac{P}{M}\left\|h_k\right\|^2 sin^2\theta_k}. \end{aligned} \tag{4.12}$$

The approximation is based on the assumption that the scheduled UEs are almost orthogonal. The $|e_k f_i|$ is Beta-distribution.

Based on the approximation function, the UE returns

$$g(k)^{MU} = \frac{\frac{P}{M}\left\|h_k\right\|^2\left|\widehat{h_k}\widetilde{f_k}cos\theta_k\right|^2}{1+\frac{P}{M}\left\|h_k\right\|^2 sin^2\theta_k}. \tag{4.13}$$

And then, the base stations calculate the SINR based on the quantized $g(k)$. $\mathbb{M}$ is a mapping function to transfer the SINR to CQI; $\mathbb{M}^{-1}$ is a mapping function to transfer the CQI to SINR. The CQI_k^{MU} is denoted by CQI of k user while

doi:10.6342/NTU201900453

considering multiple beams. It can be formulated as,

$$CQI_k^{MU} = \mathbb{M}(g(k)^{MU}).$$

The $SINR_k^{MU}$ is denoted by SINR of k user while considering multiple beams. It can be formulated as

$$SINR_k^{MU} = \frac{M}{|S| \left\| f_k \right\|^2} \mathbb{M}^{-1}(CQI_k^{MU}). \tag{4.14}$$

The overall diagram is shown in Fig. 27.

That is to say, while exploiting MU-MIMO with VIENNA, the users return the CQI considering the impact of the other beams. This way prevents the scheduler from overestimating the expected performance when scheduling multi-users in the same RB. Unfortunately, it may deteriorate the performance under the single-beam case. With the fixed mapping method, it might have no choice but to accept this drawback. However, the proposed method in the thesis allows the assigned MCS to change dynamically. As a result, we adopt the feedback, which neglects the impact of the other beams, called SU-feedback as our estimated feedback. The SU-feedback can be written as

$$g_k^{SU} = \left\| h_k \right\|^2 \left| \widehat{h_k} \widetilde{f_k} cos\theta_k + e_k \widetilde{f_k} \right|^2. \tag{4.15}$$

The CQI_k^{SU} is denoted by CQI of k user while considering single beams. It can be formulated as

$$CQI_k^{SU} = \mathbb{M}(g(k)^{SU}).$$

The estimation of SINR of user k in single-beam case is

$$SINR_k^{SU} = P\mathbb{M}^{-1}(CQI_k^{SU}). \tag{4.16}$$

By contrast, the feedback considering inter-beam interference called MU-feedback, which is represented as Eq. (4.13).

With the proposed adaption algorithm, the overall diagram is demonstrated in Fig. 28. Under the single-beam case, the scheduler uses the estimation of SINR directly.

The distribution of SU-feedback to real CQI, which including MU cases, is shown in Fig. 29. It is noticing that the pattern is different from Fig. 16. Hence, it is expected that the strategy of selecting MCS should be different. That is, the traditional method should be re-designed by researchers while the proposed method is able to be applied directly to find out a good strategy.

As for the other suggestion, it is about the retransmissions constraints of the scheduler. The number of retransmissions provides the scheduler with better capability of controlling the requirements of the communication. The benefit and impact of the constraints are shown in Fig. 54.

doi:10.6342/NTU201900453

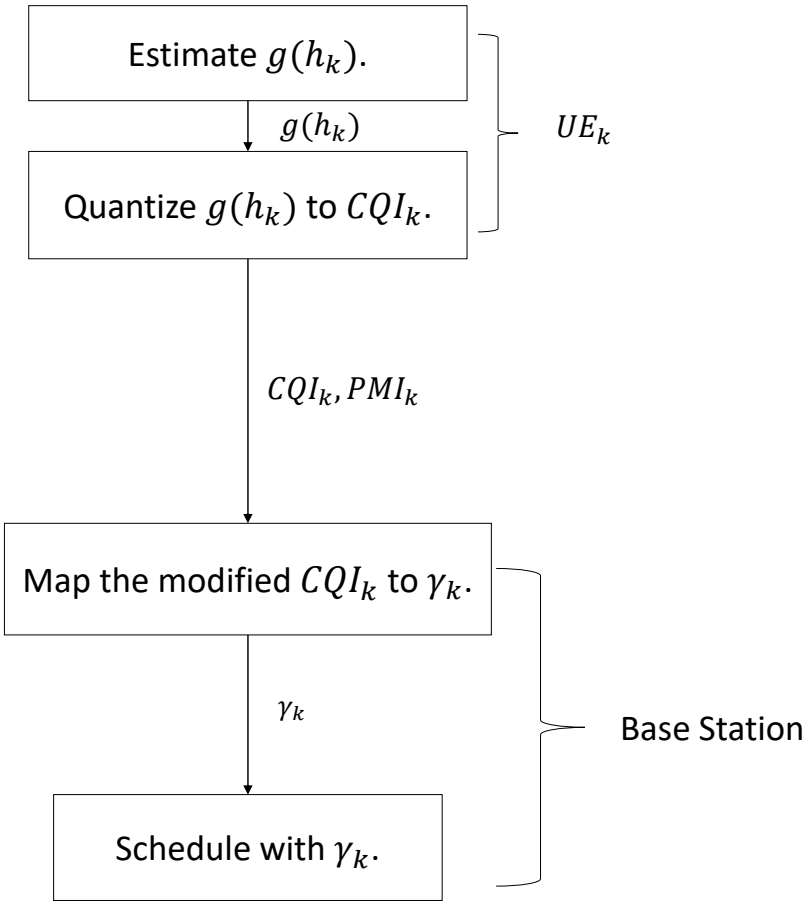


Figure 27: Diagram in VIENNA

doi:10.6342/NTU201900453

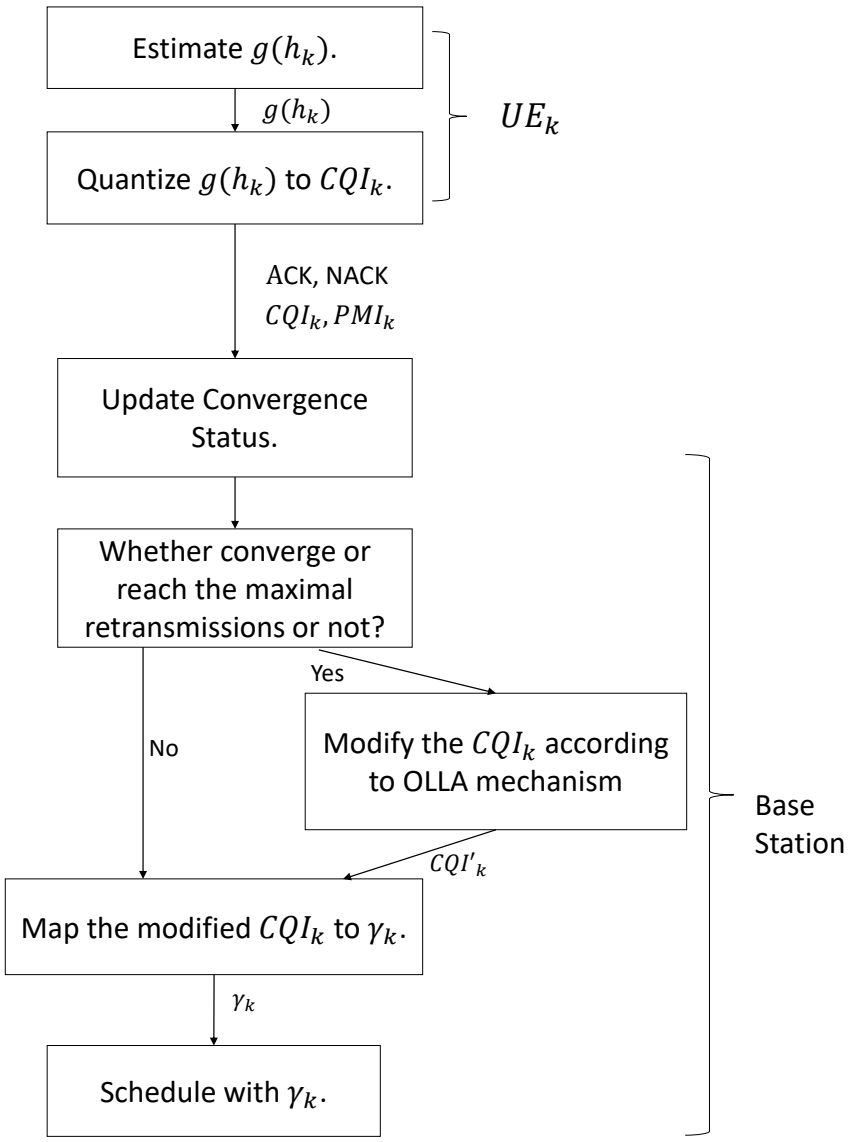


Figure 28: Modified Diagram in VIENNA

doi:10.6342/NTU201900453

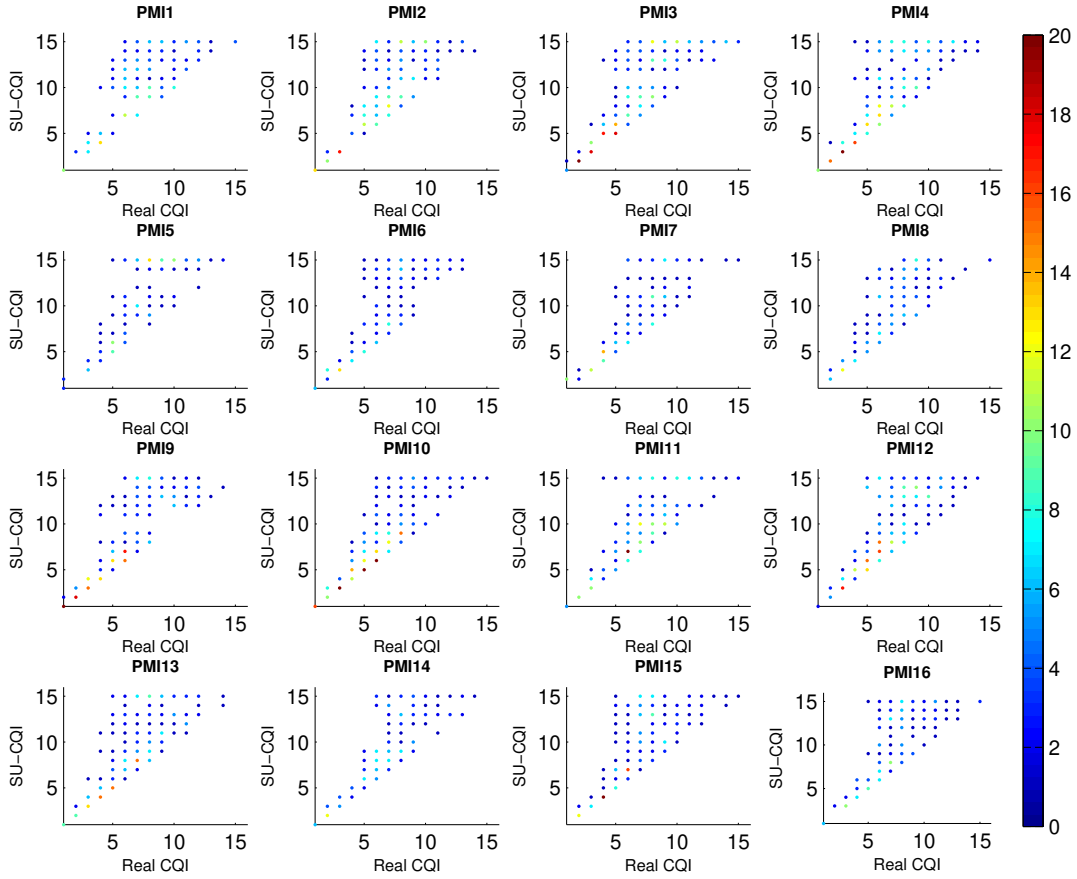


Figure 29: Real CQI and estimated CQI. Real CQI is calculated with channel h, the estimated CQI is the CQI return by UE. The estimated CQI is calculated following Eq. (4.15), which consider merely the SU-MIMO.

doi:10.6342/NTU201900453

The original scheduler chooses the user depending on the PF metric. It may good for the performance, but it results in some unpredictable results in this framework when applying OLLA mechanism. Thus, we implement the scheduler called converge-first scheduler. The diagram of the scheduler is shown in Fig. 30. This scheduler will make sure that each user has found the suitable MCS for its pairs, and then activates the original scheduler. It helps us to observe the impact of convergence steps on performance more easier. The performance of implementing the scheduler in VIENNA is shown in Fig. 53.

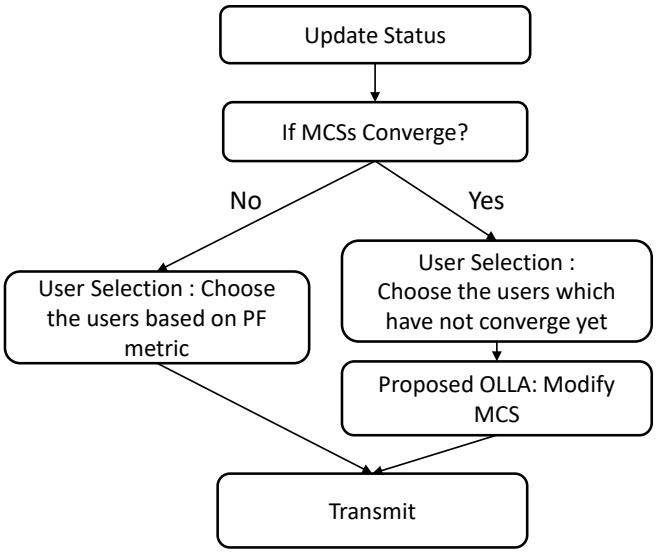


Figure 30: Converge-first Scheduler

Furthermore, there is a constraint that the beams have to be orthogonal to each others. Previous research has shown that the correlation of the beams has impact on the performance. The higher correlation usually brings the higher performance. Fig. 31 shows that adopting the constraint do not deteriorate the performance a lot. The benefit of using this constraint is that it can saving time for calculating the metrics for different grouping. Also, the N_{comb} can significantly reduced. The state space can be smaller; it is beneficial for the training. The orthogonal table is shown in Table 5.If the PMI of the user is in the left column, the PMI of the other user will be orthogonal under the condition that the PMI of the other user belong to the right column. It can be seen that the maximal number of possible paring PMI set for a PMI is 5 with this constraint, without this constraint the maximal number of the set will be the size of the codebook.

doi:10.6342/NTU201900453

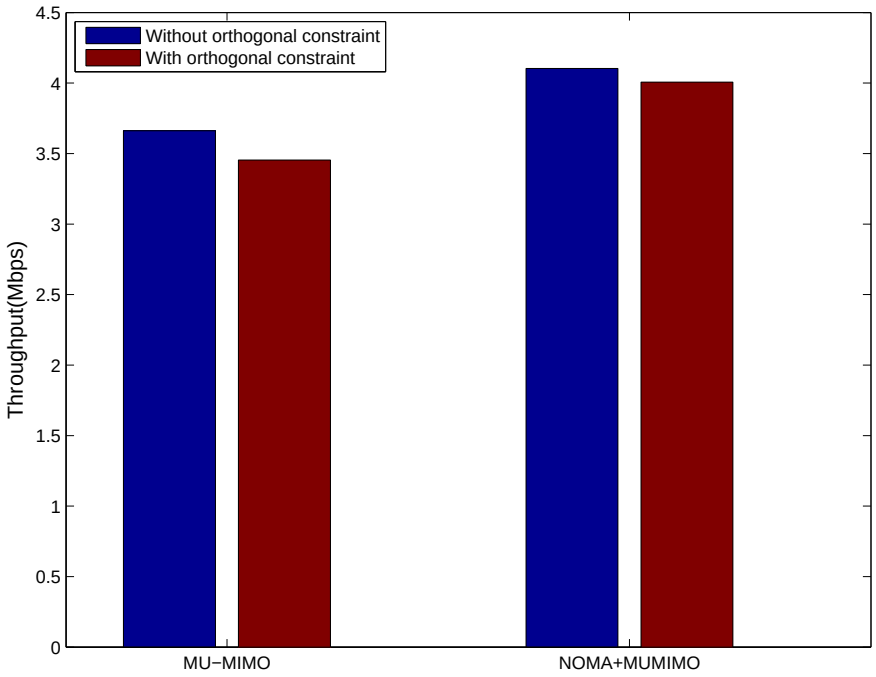


Figure 31: With orthogonal Constraint or not

Table 5: PMI Orthogonal Table

PMI	Orthogonal PMI set
1	2 3 4 9 11
2	1 3 4 10 12
3	1 2 4 9 11
4	1 2 3 10 12
5	6 7 8
6	5 7 8
7	5 6 8
8	5 6 7
9	1 3 10 11 12
10	2 4 9 11 12
11	1 3 9 11 12
12	2 4 9 10 12
13	14 15 16 10 11
14	13 15 16
15	13 14 16
16	13 14 15

doi:10.6342/NTU201900453

CHAPTER 5

PERFORMANCE EVALUATION

In this chapter, the evaluation of the proposed methods is presented. Firstly, the simulation settings are introduced. And then, we demonstrate the performance of the different structures of the neural networks, features, and exploration rules. And, we evaluate the performance of the proposed algorithm in comparison with other methods in VIENNA [21].

5.1 Scenario Setting

In order to simulate the communication as practical as possible, we adopt the Vienna LTE-A link level simulator. The simulator follows the standard of LTE. The scheduler in VIENNA follows the SNR mapping to assign proper MCS in order to prevent the receiver failing to decode the signal due to the awful channel condition. The mapping used in VIENNA is shown in Table 6. We also extend the transmitter and receiver to implement NOMA in VIENNA. The other relevant settings are shown in Table 7.

Table 6: CQI Parameters

CQI	MCS (Rate is for every 1024 bits)	Efficiency(bit/RE)
1	QPSK-78	0.1523
2	QPSK-120	0.2344
3	QPSK-193	0.3770
4	QPSK-308	0.6016
5	QPSK-449	0.8770
6	QPSK-602	1.1758
7	16QAM-378	1.4766
8	16QAM-490	1.9141
9	16QAM-616	2.4063
10	64QAM-466	2.7305
11	64QAM-567	3.3223
12	64QAM-666	3.9023
13	64QAM-772	4.5236
14	64QAM-873	5.1152
15	64QAM-948	5.5547
0(20)	0QAM-0	-10

doi:10.6342/NTU201900453

Table 7: Simulation Setting

Parameter	Value
Transmitter	1 BS with 4 antennas
Receiver	Each with 1 antenna
Carrier frequency	2.1 GHz
Bandwidth	1.4 MHz(6 RBs)
Subcarriers bandwidth	15 kHz
Number of UE	20
Channel	Temporally correlated frequency Flat Rayleigh block fading
Transmission mode	8
Beamformer	Zero forcing (ZF) and LTE codebook
Detection	Minimum mean square error (MMSE)
Feedback granularity of PMI	Whole band
Feedback granularity of CQI	1 RB
Simulation time	500 TTIs
Scheduler	PF scheduler

5.2 Simulation Results

In this section, we first the design of reinforcement learning, which are illustrated in Chapter 4. And then, we will demonstrate the advantages of the proposed method. The performance in terms of throughput, BLER, geometric mean throughput, and cell-edge user throughput in a practical communication environment with different methods are shown with VIENNA in order to put practical communication environment into consideration. The advantage of the proposed feedback and the impact of the constraint of the retransmissions will be demonstrated.

Table 8: Training Setting

Parameter	Value
ϵ_{end}	0.1
ϵ_0	0.3
γ	0.99
T	2000000
minimal batch size	32
RMSprop learning rate	0.05
c_v	0.5
c_{reg}	0.01

doi:10.6342/NTU201900453

5.2.1 Verify the Design of Reinforcement Learning

5.2.1.1 Comparison of Features

Table 9: List of different Design of State in Fig. 32

S1:

$$s_{i,t} = \left[\left[maxAckMCS \quad minNackMCS \right]_{i,t}^{N_{comb}} \quad Pair_{i,j} \quad CQI_{i,t} \quad PMI_{i,t} \right]$$

S2:

$$s_{i,t} = \left[\left[\text{ACK} \quad \text{NACK} \quad \text{not used} \right]_{i,t}^{N_{comb} \times N_{MCS}} \quad Pair_{i,j} \quad CQI_{i,t} \quad PMI_{i,t} \right]$$

S3:

$$s_{i,t} = \left[\left[maxAckMCS \quad minNackMCS \right]_{i,t}^{N_{comb}} \quad Pair_{i,j} \quad CQI_{i,t} \right]$$

S4:

$$s_{i,t} = \left[\left[maxAckMCS \quad minNackMCS \right]_{i,t}^{N_{comb}} \quad Pair_{i,j} \quad PMI_{i,t} \right]$$

S5:

$$s_{i,t} = \left[\left[maxAckMCS \quad minNackMCS \right]_{i,t}^{N_{comb}} \quad Pair_{i,j} \quad CQI_{i,t} \quad PMI_{i,t} \right]$$

(without one hot encoder)

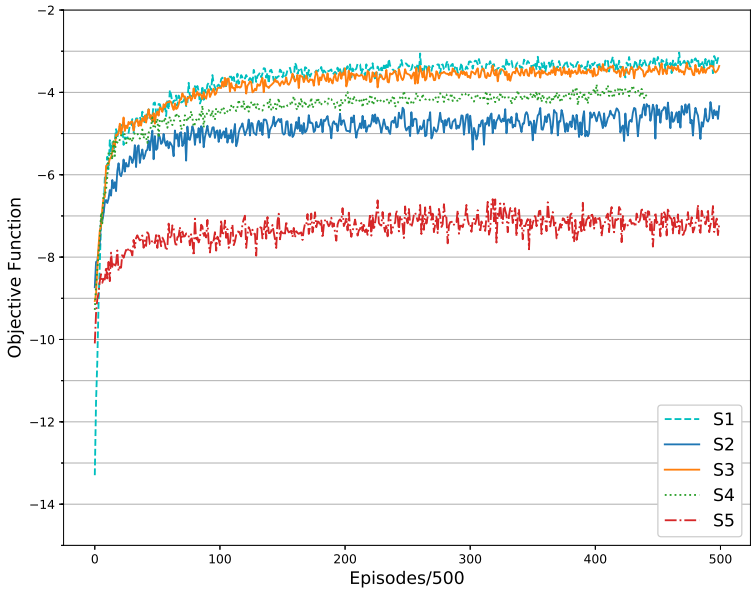


Figure 32: Comparison of States

The effectiveness of each feature and feature simplification is shown in Fig. 32.

doi:10.6342/NTU201900453

Different features are disabled in a different state in order to verify the effectiveness of the absent feature.

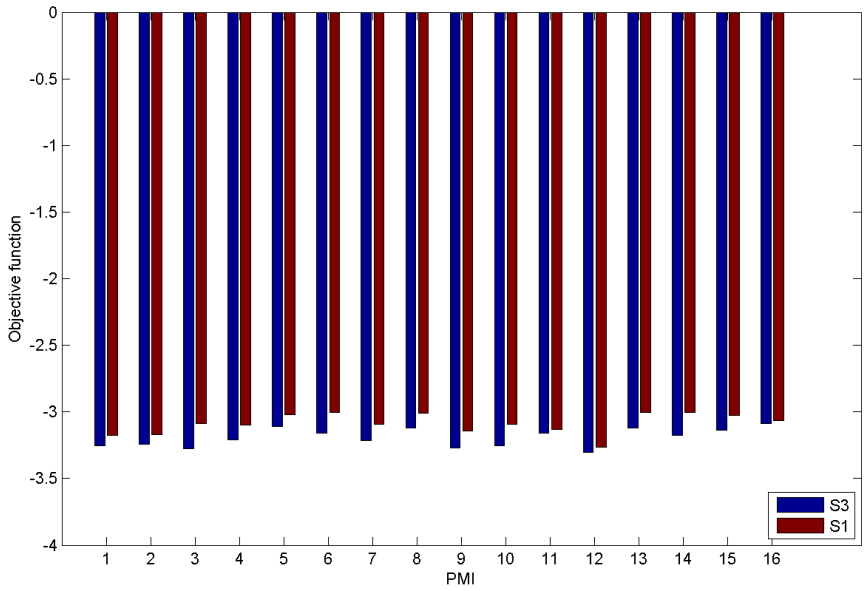


Figure 33: Comparison for each PMI with S1 and S3

Taking a look to S3, in spite of the slight improvement, it still can say that considering PMI of the user does not only benefit the architecture of the neural network but also the states. It can be seen in Fig. 33 that the objective function with S1 is better than with S3. S1 shows the capability of capturing the characteristics of each PMI. As for S2, it can be said that the simplification is very effective because the training speed is the worst, and the variation in the training process is large. As for S4, it implies that the CQI is a feature which is more important than the PMI of the user. S5 demonstrates the importance of one hot encoder. It seems that without proper transform the performance will be much worse.

Unlike traditional methods usually only consider NACK and ACK, or limited information. This result shows that the proposed method can handle more parameters in the environment. Also, if the simplification does not lose the most important information, it will show a positive result.

5.2.1.2 Separate or Fully-connected

In Chapter 3, we learned that different PMIs have distinct patterns of the distribution of the estimated SINR. Based on this observation, we separate the neural network for each PMI, which is demonstrated in Fig. 22. We hypothesize that the separation can help the neural network to learn the strategy for PMI more efficiently and to capture the different characteristic for each PMI more easily.

doi:10.6342/NTU201900453

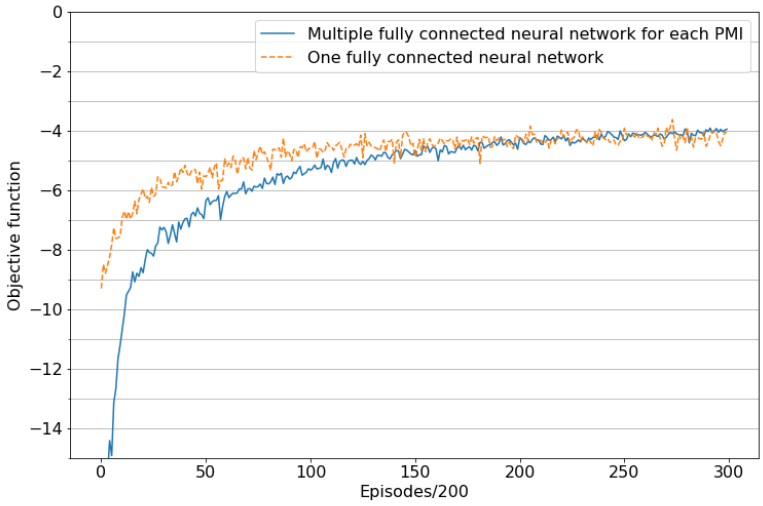


Figure 34: Training Speed for one fully-connected Network and multiple fully-connected Network for each PMI

Fig. 34 compares the training efficiency between the one fully-connected network and multiple fully-connected networks for each PMI. The multiple fully-connected networks are trained separately, so the objective function of the multiple fully-connected networks in Fig. 34 is the sum of the objective function of each network for each episode. It seems that one fully-connected network have better performance in the beginning, but the separated network surpasses the one fully-connected network in the end. It might be that some of the characteristics of the PMI are not easy to learn, resulting in the performance loss in the beginning. Nevertheless, the separated networks still show higher potential to carefully catch the characteristics for each PMI, leading to a higher performance in the end. These results suggest that if multiple computers are available to train the proposed architecture in parallel, this architecture can save us time and learn better. Still, we can obtain a decent result with one fully-connected neural network. Due to the acceptable performance of the one fully-connected network and lacking multiple hardware, we decide to apply one fully-connected neural network. Still, this result suggests that the task in this thesis can benefit from the separated design for distinct PMI.

5.2.1.3 Neurons and Layers

Despite that there are several rules of thumb about the choice of the layers and neurons in the neural network, a clear theory is still absent. In practical, validating the choice in the neural network in order to obtain a good-trained neural

doi:10.6342/NTU201900453

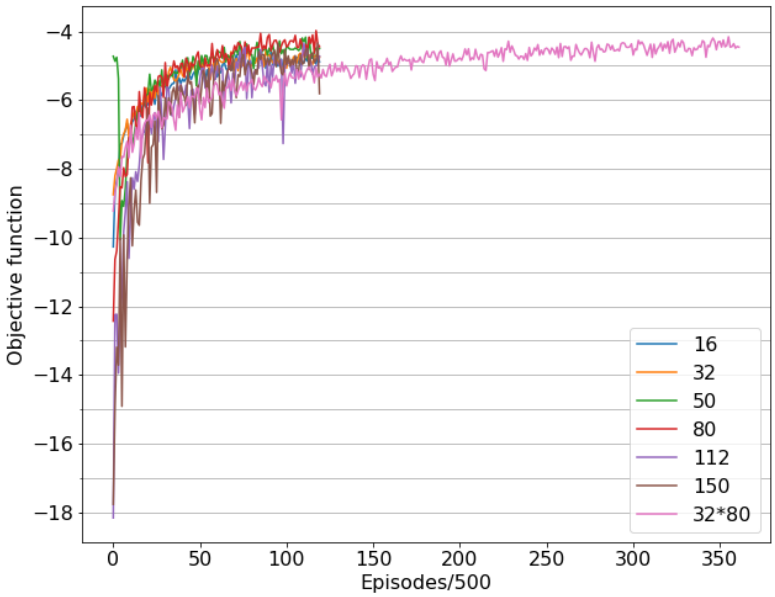


Figure 35: Training speed of different Neurons and Layers

Table 10: List of Convergence Steps for different Number of Neurons

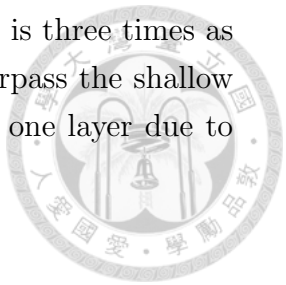
Number of Neurons in Layer 1	Average Steps
16	4.26
32	4.16
50	4.26
80	4.2
112	4.87
150	4.59

network is necessary. From Fig. 35 there is no significant difference between the number of neurons with one layer, so we compare the trained agents in the realistic environment in Table 10. It seems that the number of neurons = 32 is better.

Researches have studied on the benefit of the deep and shallow neural network. The deep neural network may perform better than shallow neural network while the number of parameters is the same in the neural network because the deep neural network can perform a hierarchical structure. However, the previous researches also point out that deep neural learning is very difficult to train well to achieve the expected result. In most cases, the neural network with one layer is

doi:10.6342/NTU201900453

already good enough. In Fig. 35, the training time of two layers is three times as much as of one layer. Since the deep neural network did not surpass the shallow neural network significantly, we adopt the neural network with one layer due to the faster training speed.



5.2.1.4 The rules of exploration and exploitation

Algorithm 3 Rule 1: Select Actions with Constraint in ϵ-greedy

\\ ϵ_0 is the initial value of ϵ; ϵ_{end} is the end value of ϵ; ϵ_t is the value of ϵ at step t \\ t is current number of training step; ϵ_0 will reach ϵ_{end} after T steps $t < T$ $\epsilon_{t+1} = \epsilon_t - (\epsilon_0 - \epsilon_{end})/T$ $\epsilon_{t+1} = \epsilon_{end}$ i =random number picked from [0,1] $i < \epsilon_{t+1}$ a_{t+1} picked randomly from [($maxAckMCS$+1),($minNackMCS$-1)] a_{t+1} selected according to the policy

Algorithm 4 Rule 2: Select Actions without constraint in ϵ-greedy

\\ ϵ_0 is the initial value of ϵ; ϵ_{end} is the end value of ϵ; ϵ_t is the value of ϵ at step t \\ t is current number of training step; ϵ_0 will reach ϵ_{end} after T steps $t < T$ $\epsilon_{t+1} = \epsilon_t - (\epsilon_0 - \epsilon_{end})/T$ $\epsilon_{t+1} = \epsilon_{end}$ i =random number picked from [0,1] $i < \epsilon_{t+1}$ MCS picked randomly within $\mathcal{M}$ a_{t+1} selected according to the policy

Algorithm 5 Rule 3: Select Actions based on the policy without ϵ-greedy

a_{t+1} selected according to the policy

As Chapter 4 said, Algorithm 3(rule 1) is designed for learning more efficiently by confining the exploration space in a reasonable way. The Algorithm 5 is the general algorithm for a policy-based method. We compare these three strategies of exploration and exploitation in Fig. 36, it can be seen that confining the action space can accelerate the learning speed. Also, if the exploration and exploitation only rely on the stochastic policy, the selection of action might be biased by the beginning value of the policy. That is, the action space cannot be well explored. Also, it is noticing that the learning process is more stable with Algorithm 3(rule 1) than with Algorithm 4(rule2) and Algorithm 5(rule3).

The result implies that while considering exploration and exploitation, confining the exploration space based on the domain knowledge is useful while the exploration space is large.

5.2.1.5 N and R

Reward shaping is a crucial issue to guide the training agent to learn the desired behavior, the reward function is shown as

doi:10.6342/NTU201900453

Figure 36: Comparison of Exploration Rules

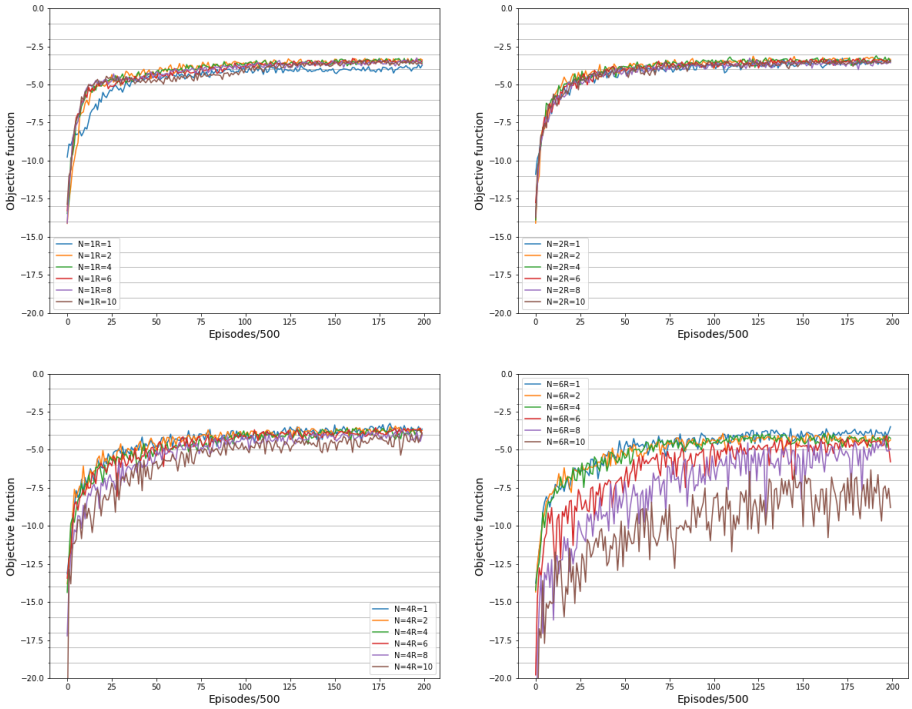


Figure 37: Training speed with different N and R

doi:10.6342/NTU201900453

$$r_{i,t} = \begin{cases} -R, & \text{if } a_{i,t} < maxAckMCS_{i,t}^{Pair_{i,j}} \text{ or } a_{i,t} > minNackMCS_{i,t}^{Pair_{i,j}} \\ 0, & \text{else if } |maxAckMCS_{i,t}^{Pair_{i,j}} - minNackMCS_{i,t}^{Pair_{i,j}}| \leqslant 1 \\ & \text{and } a_{i,t} == maxAckMCS_{i,t}^{Pair_{i,j}} \\ -1, & otherwise \end{cases} . \tag{5.1}$$

N-step return may improve the performance due to larger reward in an iteration. According to the previous study, both the magnitude of the N and R have to be carefully chosen. A large value will cause high variance and divergence, a small value cannot encourage or discourage the network. Since both magnitudes have an impact on the performance, we put them together to validate which combination is better.

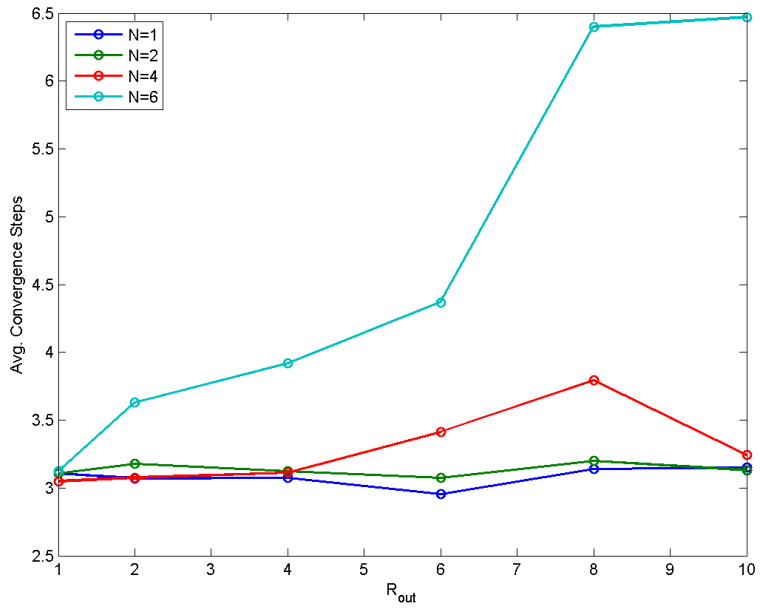


Figure 38: Convergence Steps with different Parameters

In Fig. 37, it can be observed that as $N = 4, 6$ the variance is larger when the R is higher. Especially as $N = 6$, it seems that the policy cannot converge if R is too large. It can be said that the agent cannot benefit for the R if N is large. This result is reasonable, if both R and N are large at the same time, the cumulative reward might be extremely large. And, the policy may suffer from the higher variance of the reward, which may cause diverge. On the condition that $N = 1$, the agent learns slower if $R = 1$. If $R > 1$, the agent can learn faster. The difference between $R = 2 - 10$ is not very obvious. Also, if $N = 2$, the value of R has no obvious impact according to Fig. 37.

doi:10.6342/NTU201900453

Since the best performance is not very clear in Fig. 37 while considering training speed, we compare the convergence steps. In Fig. 38, the value between $N = 1$ and $N = 2$ is close, so it is still hard to tell which NR setting is the best. However, we choose $N = 1, R = 6$ in this thesis because it seems slightly better than the others in these trained agents.

From this result, we can conclude that the optimal R is dependent on the chosen of N. In this task, the chosen of N is more important. If chosen N is within a certain range, the R will have a fewer impact.

- **N, R, and the rules of exploration and exploitation**

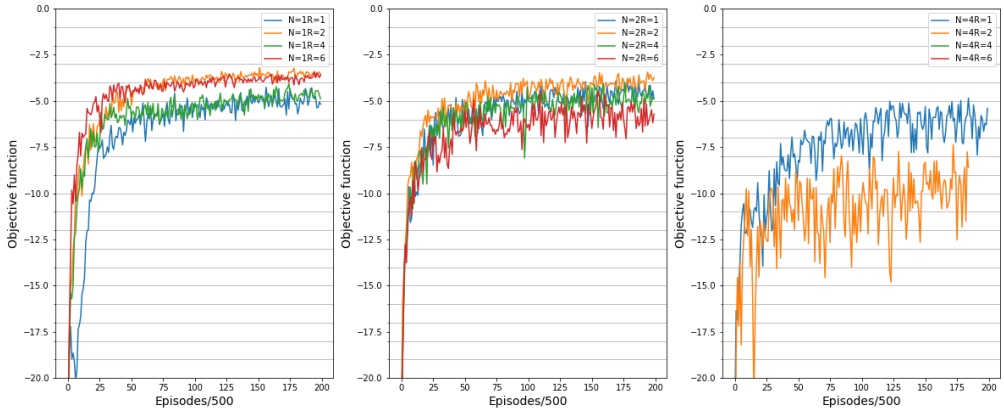


Figure 39: Training Speed with different N and R with Rule 2

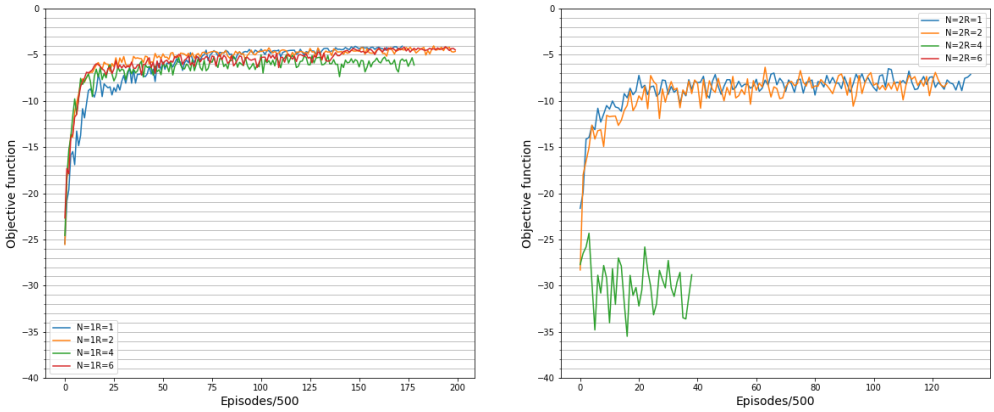


Figure 40: Training Speed with different N and R without ϵ-greedy Algorithm

doi:10.6342/NTU201900453

Since both R and Algorithm 3 are designed based on the knowledge of the effective action range, we would like to observe how the rule affect the performance while validating N and R.

It can be seen that the variance in Fig. 39 is higher than in Fig. 38. Also, unlike in Fig. 38, we can conclude that the chosen of N is less relevant with R when N is small. It seems that without confining the action space, the encouragement and discouragement of reward plays a more important rule. Furthermore, the variance is increased positively with N.

This trend is ever significant in Fig. 39, the performance of very sensitive to the N and R. The reason that $N = 2, R = 6$ vanishes in the figure is that the value is to exceed the range of the y-axis.$N = 2, R = 6$ is shorter because the agent learns slow, it spends more time in each episode. Thus, with the same training time, it can only operate fewer episodes.

In short, restricting the action space is more effective than the rewarding, leading to a much more smoother learning curve. Furthermore, restricting the action space allows us to spend less effort for chosen N and R. Still, reward shows much more benefits when action space is unknown, but the tendency of the desired behavior is known, such as R_{Nack} in sec 5.2.2.6. Also, the chosen of N is important.

5.2.2 Performance in VIENNA

In this section, we demonstrates how we choose the parameters in the **Proposed OLLA**, **Traditional OLLA** [9], and **Baseline OLLA** [15]. Also, we compare the performance of the **Original**, which is a method applying in VIENNA without OLLA and with limited CSI; **Traditional OLLA**, which is the first OLLA; **Baseline OLLA**, which is enhanced OLLA; Proposed method; Perfect, which is without and with perfect CSI. Furthermore, the benefit of the proposed feedback is demonstrated in this section.

We consider 4 metrics: Throughput, BLER, Geometric mean throughput, and cell-edge user throughput. Throughput is the data that the base station can send within a certain time. BLER is the ratio of the number of erroneous transport blocks and the transmitted transport blocks. The geometric mean rate is the product of the average throughput of all users, written as

$$\sqrt[n]{\prod_{u \in \mathcal{U}} \overline{R_u}},$$

where $\overline{R_u}$ is the average throughput of user$_u$. By definition, the cell-edge user throughput the 5th percentile point of the CDF of user throughput. In the thesis, it represents the throughput of the last user because the number of UE in the simulation is 20.

doi:10.6342/NTU201900453

5.2.2.1 Traditional Method

The traditional method changes the MCS in a fixed step. If the base station receives a NACK, the next MCS will be decreased by Δ_{down}; if the base station receives an ACK, the next MCS will be increased by 1. The mechanism can be written as

$$MCS_{t+1} = \begin{cases} MCS_t - \Delta_{down} & \text{, if receiving an NACK} \\ MCS_t + 1 & \text{, if receiving an ACK} \end{cases} \tag{5.2}$$

The ratio of NACK is defined as

$$\text{The ratio of NACK} = \frac{\text{Total Nack steps}}{\text{Total convergence steps}}.$$

The convergence steps and the ratio of NACK both have an impact on the

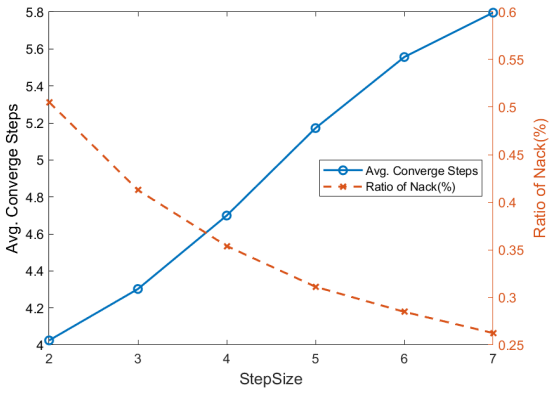


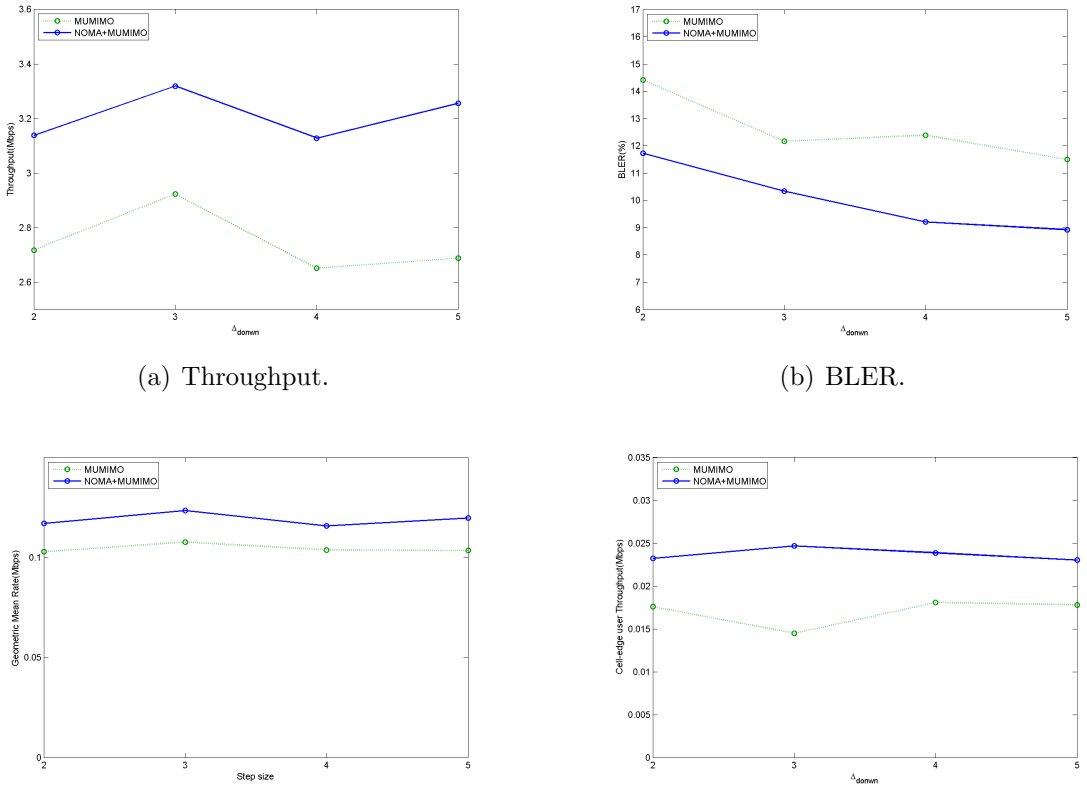
Figure 41: Relationship between Step Size, Convergence Steps, and Ratio of Nack

throughput. The convergence steps represent the capability of recovering from the sudden change in the channel. If the chosen MCS can find the suitable MCS as fast as possible, the performance could recover faster. However, the searching process may deteriorate the performance. If the chosen steps are too aggressive, it may cause failed transmission and sacrifice the throughput. Thus, observing these two metrics can help us to predict the performance easier.

It can be seen in Fig 41, the larger the Δ_{down} is, the smaller the ratio of NACK. For the reason that if the Δ_{down} is larger, the probability of selecting MCS larger than the real SINR is smaller.

In Fig. 42, it can be observed that while Δ_{down} is increasing, the BLER is decreasing. The throughput does not decrease with BLER. As mentioned before, both convergence steps and BLER can affect this metric. Since the throughput is highest as $\Delta_{down} = 3$, we choose this value in this thesis for the traditional method.

doi:10.6342/NTU201900453



(a) Throughput.

(b) BLER.

(c) Geometric mean rate.

(d) Cell-edge throughput.

Figure 42: Performance in Traditional Method

5.2.2.2 Baseline Method

The baseline method changes the Δ_{down} and Δ_{down} according to elapsed time [15].

$$MCS_{t+1} = \begin{cases} MCS_t - \Delta_{down,t} & \text{, if receiving an NACK} \\ MCS_t + \Delta_{up,t} & \text{, if receiving an ACK} \end{cases}, \tag{5.3}$$

where $\Delta_{down,t} = A_{\text{Offset}} + A_{\text{Initial}} \cdot \exp^{-\gamma t}$, $\Delta_{up,t} = a(A_{\text{Offset}} + A_{\text{Initial}})$.

There are several parameters in the baseline. A_{Offset} is the offset value, $\Delta_{down,t}$ will converge to A_{Offset} in the end. We compare the parameters of the baseline; A_{Initial} has the impact in the beginning; γ determined the rate of $\Delta_{down,t}$ converging to A_{Offset}; a is the ratio between $\Delta_{down,t}$ and $\Delta_{up,t}$. In Fig. 43, the γ has smaller impact on the performance. If $A_{\text{Initial}} = 0$, its behavior is the same as traditional method. On the condition that $A_{offset} = 0$, the performance looks better. We choose $A_{\text{Initial}} = 1$, $gamma = 1$, and $A_{\text{offset}} = 2$ in the end. The convergence steps is not the smallest in this setting, but it only sacrifices the convergence steps and improves the ratio of NACK a lot.

Fig. 44 shows how the baseline works with the chosen setting. The first Δ_{down} is 3 and then become 2. The Δ_{up} is always 1. The step size is varied with time.

doi:10.6342/NTU201900453

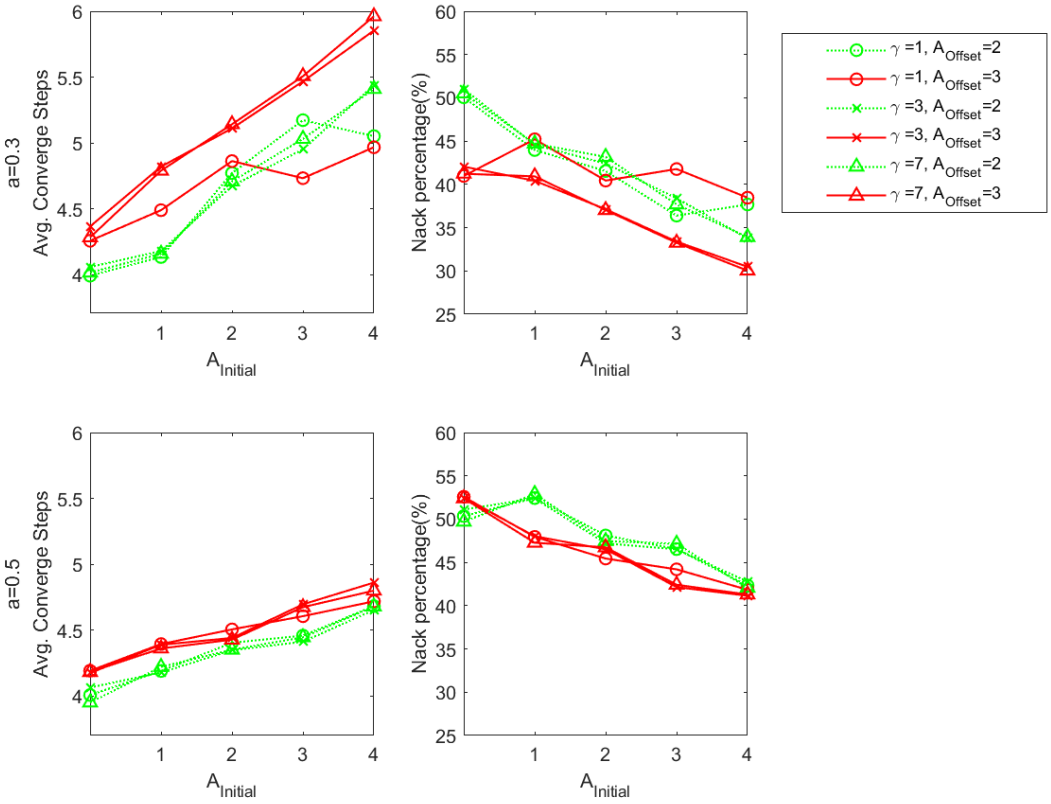


Figure 43: Comparison of different Parameters of the Baseline

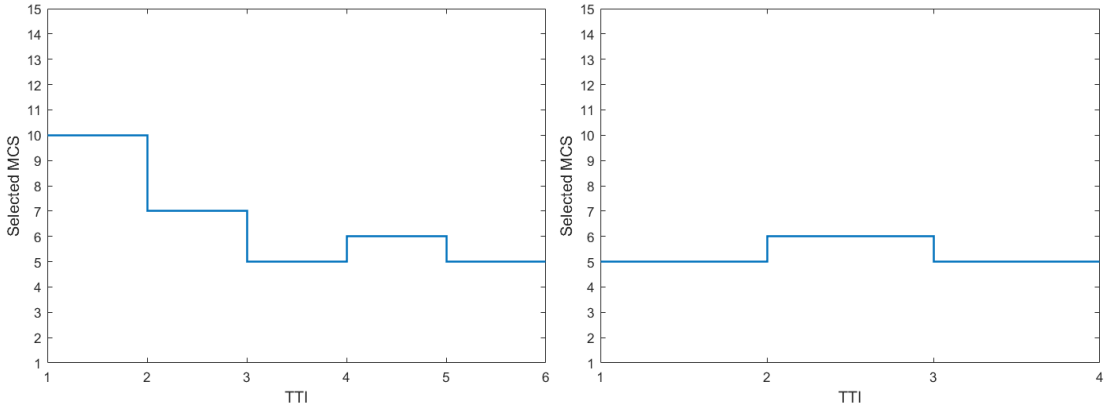


Figure 44: Demonstrate how the chosen MCS changes while $A_{\text{Initial}} = 1$, $gamma = 1$, and $A_{\text{offset}} = 2$

doi:10.6342/NTU201900453

5.2.2.3 *Comparison of Convergence Steps between different Methods in different Types of Feedbacks*

The convergences steps is defined as -objective function. The formula of convergences steps can be written as

$$mean\left[\sum_{t=0}^{T} r_t\right]$$

$$r_t = \begin{cases} 0, & \text{if the base station knows that it has reached the suitale MCS.} \\ 1, & \text{otherwise.} \end{cases} . \tag{5.4}$$

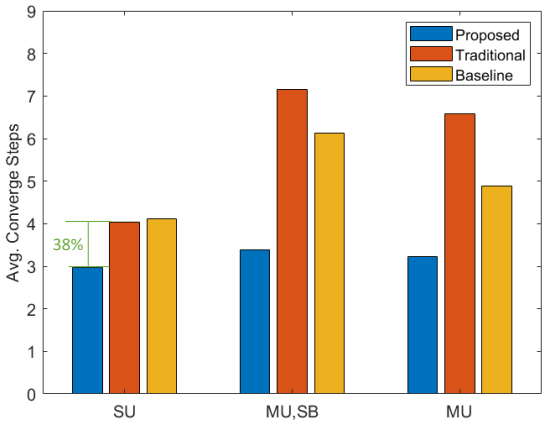


Figure 45: Comparison of Convergence Steps between different Methods in different types of Feedbacks

SU indicates that OLLA exploits SUMIMO feedback; **MU** indicates that OLLA exploits MUMIMO feedback, and is not activated in single beam case; **MU, SB** indicates that OLLA exploits MUMIMO feedback, and is activated in single beam case. The reason that **SU** do not need to consider OLLA in single beam case is that it can use SUMIMO feedback directly.

Fig. 45 shows the convergence speed in each OLLA methods. All the methods aim to converge as fast as possible. It can be observed that the proposed method shows the stronger capability in finding a good strategy of converging fast and dealing with the different condition in the channel.

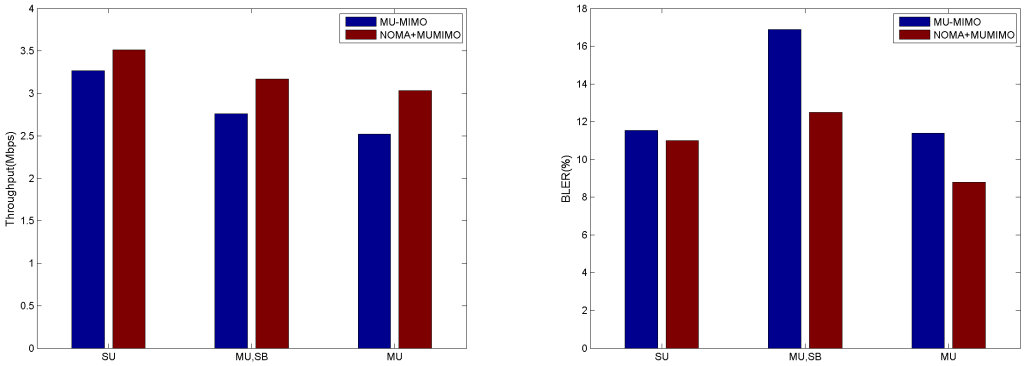
5.2.2.4 *Performance in different Types of Feedbacks*

Fig. 46 explains that why the SU-feedback is adopted in this work. Notice the throughput of SU and MU, SB; the throughput is higher if we considering the capacity of single beam case. Also, directly using SU-feedback is better, because

doi:10.6342/NTU201900453

the base station does not need to spend time on searching for the suitable MCS while scheduling in single beam. Furthermore, it is noticing that the BLER is high if we use MU-MIMO feedback to do OLLA in a single beam because the base station has to spend more effort in searching for the suitable MCS for each possible grouping.

Overall, SU feedback shows better performance, so we adopt SU feedback in this thesis.



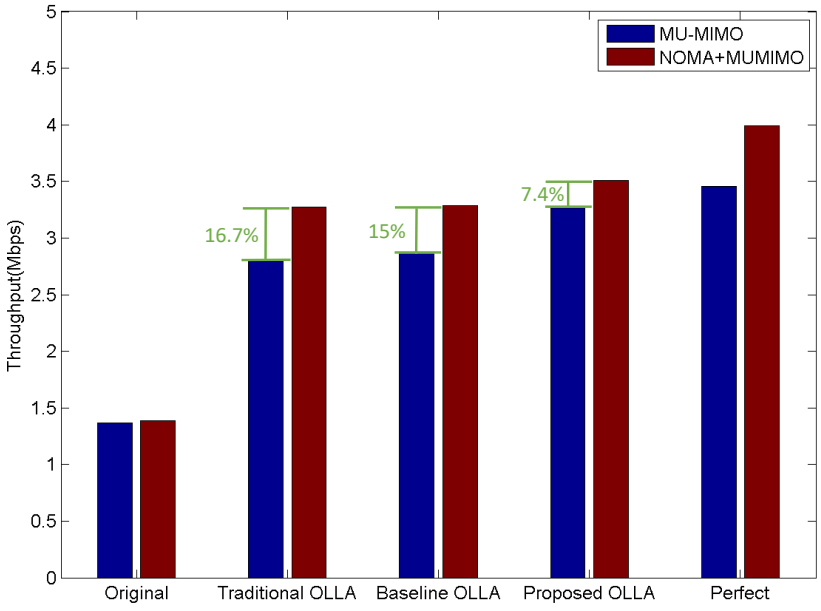
(a) Throughput. (b) BLER.

Figure 46: Performance in different Types of Feedbacks

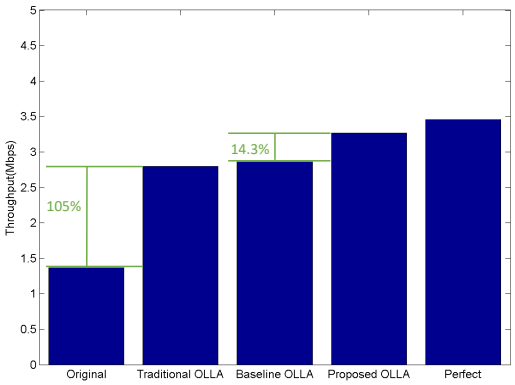
5.2.2.5 Performance in different Methods

When it comes to throughput, we can observe Fig. 47. Firstly, the throughput is double with OLLA. The traditional OLLA increases the throughput by 105% in MU-MIMO and by 137% in NOMA+MUMIMO, respectively. This result implies that increasing the accuracy of the SINR benefits the throughput significantly. Moreover, the improvement between MU-MIMO and NOMA+MUMIMO is more obvious. The gain between MU-MIMO and NOMA+MUMIMO is almost can be ignored in the original method, while the gain is 7.4% in the proposed method. Furthermore, if we compare the throughput of the proposed method with the baseline method. It increases by 14.3% and by 6.7%, in MU-MIMO and in NOMA+MUMIMO, respectively. The cell-edge user throughput can be observed in Fig. 48, it is terribly small with original method, because the MU-MIMO feedback is the lower bound of the expectation of SINR. Without OLLA, weak users can only transmit data with small MCS and have less chance to schedule with PF scheduler due to the underestimated SINR. In this situation, OLLA can improve the cell-edge user throughput a lot. Unfortunately, it does not guarantee the cell edge throughput and BLER. Because our target is to find a strategy to converge

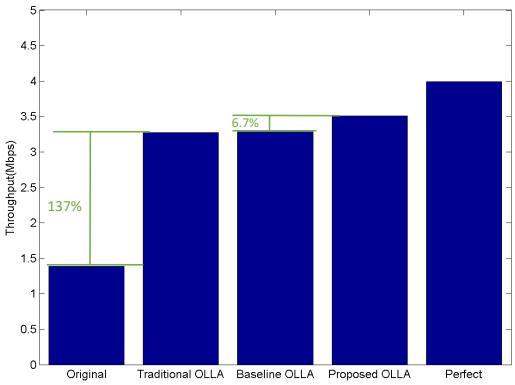
doi:10.6342/NTU201900453



(a) NOMA+MU-MIMO and MU-MIMO.

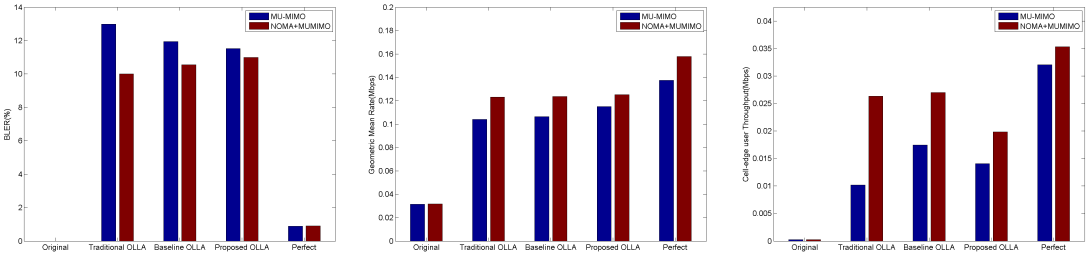


(b) MU-MIMO.



(c) NOMA+MU-MIMO.

Figure 47: Throughput in different Method



(a) BLER. (b) Geometric mean rate. (c) Cell-edge throughput.

Figure 48: Performance in different Methods

as fast as possible, this optimal strategy may be very aggressive. Fig. 49 demonstrates how each OLLA changes the selection of MCS according to the HARQ

doi:10.6342/NTU201900453

information.

Still, the proposed OLLA performs well in terms of geometric mean rate. In short, the result indicates that improving the convergence speed has a positive impact on the throughput and fairness, but do not guarantee the BLER.

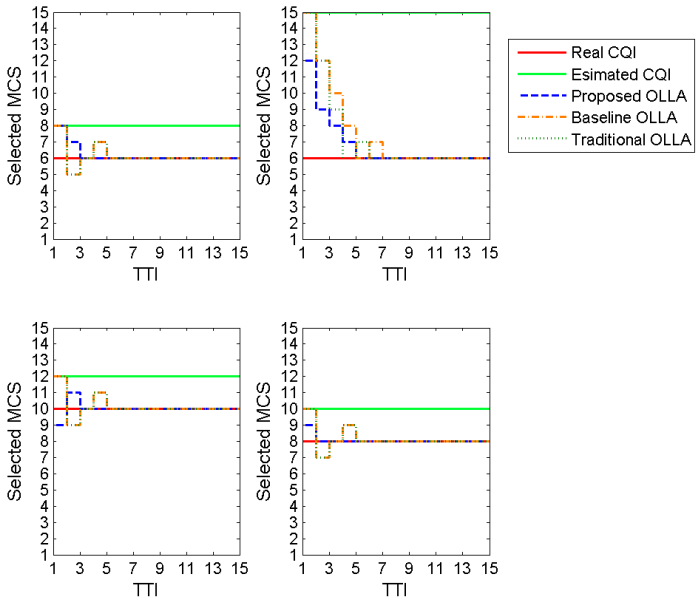


Figure 49: Demonstration for each OLLA Method

5.2.2.6 Impact of R_{Nack}

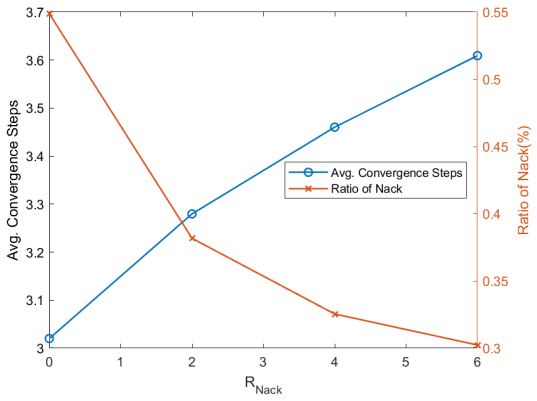


Figure 50: Relationship between R_{Nack}, Convergence Steps, and Ratio of Nack

From previous simulations, we learns that if only considering convergence may cause higher BLER. Thus, we proposed a mechanism to control the BLER. We use reward shaping to control the bahavior of the trained agent. The reward function

doi:10.6342/NTU201900453

is defined as

$$r_{i,t} = \begin{cases} -6, & \text{if } a_{i,t} < maxAckMCS_{i,t}^{Pair_{i,j}} \text{ or } a_{i,t} > minNackMCS_{i,t}^{Pair_{i,j}} \\ 0, & \text{else if } |maxAckMCS_{i,t}^{Pair_{i,j}} - minNackMCS_{i,t}^{Pair_{i,j}}| \leqslant 1 \\ & \text{and } a_{i,t} == maxAckMCS_{i,t}^{Pair_{i,j}} \\ -1, & \text{else if } a_{i,t} \leqslant realSINR_{i,j} \\ -R_{Nack}, & \text{else if } a_{i,t} > realSINR_{i,j} \end{cases} \tag{5.5}$$

In theory, R_{Nack} depicts how reluctant is the agent to choose the MCS, which has

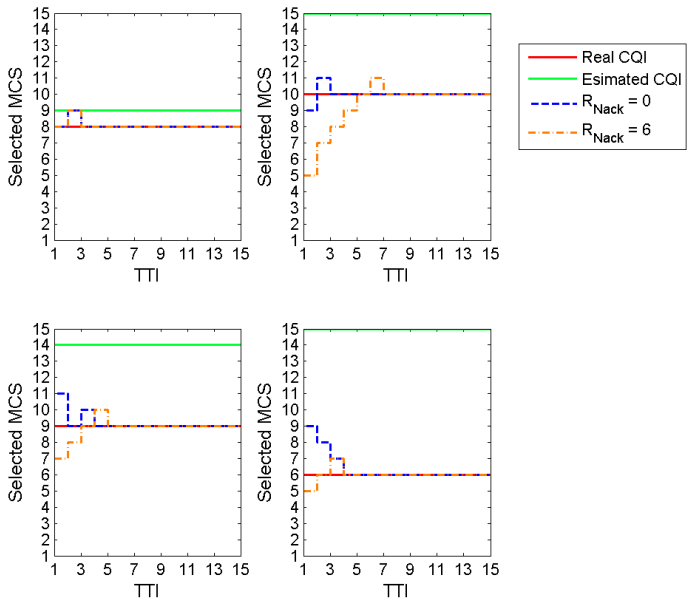


Figure 51: Demonstration of $R_{Nack} = 0$ and $R_{Nack} = 6$

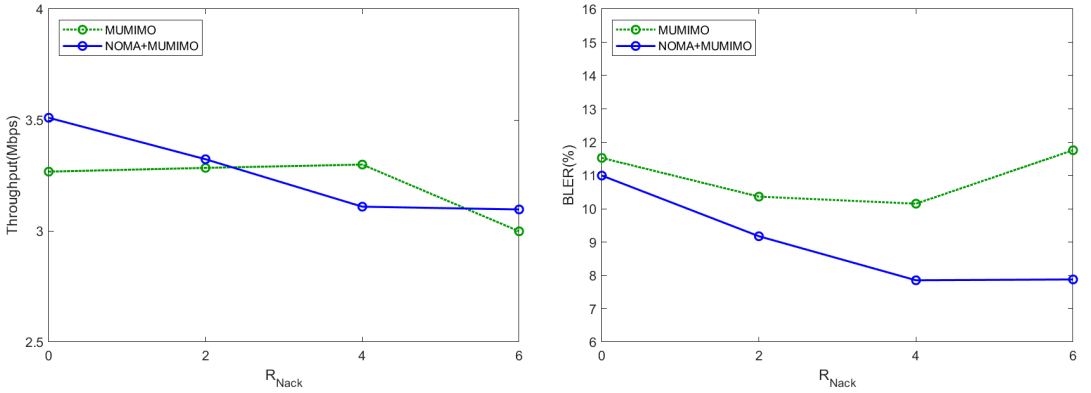
a higher probability of causing higher BLER.

Fig. 50 indicates that R_{NACK} can control the how aggressive is the chosen steps effectively.

From Fig. 51, it seems that while $R_{NACK} = 6$, the chosen MCS is very conservative.

In Fig. 52, the BLER tends to be smaller as R_{Nack} is larger. It is noticing that BLER become higher when $R_{Nack} = 6$. It may be caused by the characteristics of the PF scheduler. Because this type of scheduler considers fairness and throughput at the same time, it might tend to choose the user with aggressive estimated SINR. To verify this hypothesis, we modify the scheduler and call the new scheduler as a converge-first scheduler. In this converge-first scheduler, all the users have to find their own suitable MCS at first. After all the groupings find the suitable MCS,

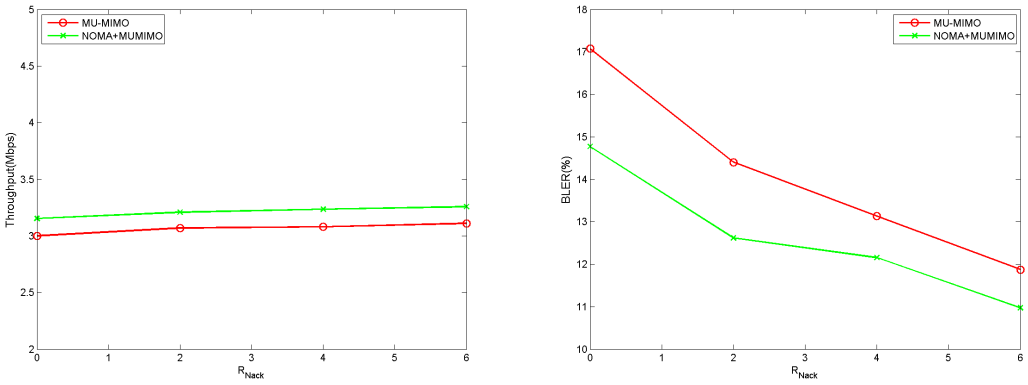
doi:10.6342/NTU201900453



(a) Throughput. (b) BLER.

Figure 52: Impact of R_{Nack} on Performance

the original PF scheduler starts. This setting can prevent the scheduler from only choosing the grouping with aggressive estimated SINR. The results are shown in Fig. 53. It can be observed that the BLER is negatively correlated with R_{Nack}. Despite that this type of the scheduler does not show the advantage in terms of throughput, it is useful for understanding the influence of the proposed method. It is noticing in Fig. 52 that the convergence subframes in NOMA+MUMIMO are less than in MU-MIMO. It implies that if more the users can be scheduled at the same time, quicker can the base station find the suitable MCS of users.



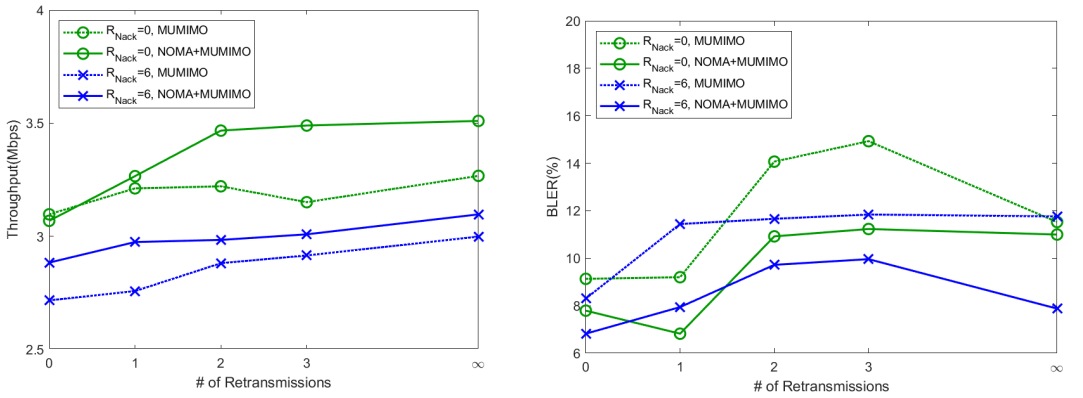
(a) Throughput. (b) BLER.

Figure 53: Impact of R_{Nack} on Performance in converge-first Scheduler

5.2.2.7 Performance of Proposed Method with Constraints of Retransmissions

We observed that the behavior of the trained agent tends to choose conservative step in the beginning steps if R_{Nack} is larger, so we hypothesise that the performance could be improved even if the base station does not find the most

doi:10.6342/NTU201900453



(a) Throughput. (b) BLER.

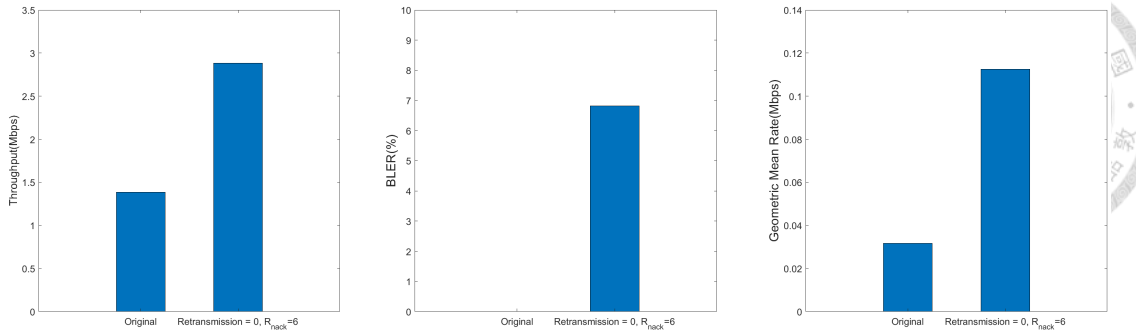
Figure 54: The trend of each metrics varies with the number of retransmissions.

suitable MCS. To verify this hypothesis, we constrain the number of the retransmissions. The impact of the limitation of the number of retransmissions and the value of R_{Nack} on the performance in Fig. 54. The impact of not finding the most suitable MCS can be observed as well. We compare the performance in two conditions: one is $R_{Nack} = 0$, the other is $R_{Nack} = 6$ because these two conditions have two different tendencies in choosing the MCS in the beginning.

In Fig. 54, it can be seen that the throughput is highest if there is no limit in retransmissions. Nevertheless, the throughput is not low while the retransmission is 0.

From the previous simulation, it can be observed that if $R_{Nack} = 0$, the chosen steps tend to be aggressive, while the chosen steps tend to be more conservative as $R_{Nack} = 6$. Thus, while the number of the retransmission=0, the $R_{Nack} = 6$ is smaller due to the more conservative choice. The reason that why the BLER is not highest while the number of retransmissions is unlimited might be is that if there is limitation, the base station will choose the maximal available assigned MCS of UE, which might be much smaller than the suitable MCS or the other UEs' modified MCS according to proposed OLLA, while the limitation of the UE is reached. Thus, the PF-scheduler has higher possibility to choose the other UE due to the higher modified but possibly too aggressive MCS and the smaller MCS of the UE, who has achieved limitation. Besides, the base station has to know which is minimal MCS within the MCSs received NACK, and the average steps, which definitely contains a NACK, is usually between 3 and 4. It implies that the UE might return at least a NACK within 4 retransmissions if it has reached the suitable MCS. However, if the suitable MCS is not achieved, the NACK might have never appeared. This is why the BLER is not correlated with the number of the retransmissions.

doi:10.6342/NTU201900453



(a) Throughput. (b) BLER. (c) Geometric mean rate.

Figure 55: Comparison between original Method and proposed Method with Constraint of Retransmission=0 in NOMA+MU-MIMO

This constraint also brings another benefit, when the number of the retransmissions is 0. No retransmissions mean that the overhead is the same as the original feedback, but the performance is better as shown in Fig. 55. The throughput is doubled; The BLER is below 10%; The cell-edge use throughput increased significantly. It implies that the initial value according to the proposed OLLA is good.

The reason that why the proposed OLLA can be explained in Fig. 21. In MU-MIMO, if the user paired with a different user, the real SINR, γ, will be different. Due to the interference, the user can only return the lower bound of the expected SINR according to the assumption of the distribution of SINR. The traditional mapping is an only one to one mapping. However, our proposed method can serve as a multiple dimensional mapping. Due to the multiple outputs, the capacity of the single user does not have to be limited, the SU-MIMO feedback can be adopted. The capacity of the multiple beams can be measured through the training process. It is noticing that the good initial value is not our designed target, but the initial value is lower could be because of the strong punishment while receiving NACK; so if the better performance is desired for this type of application, the redesign of the rewarding function is needed. Still, the design of the framework of reinforcement learning in this type of problem could remain.

In short, unlike supervised learning, the labelling is necessary, reinforcement learning may have the higher potential of searching for complicate mapping, which has higher uncertainty but the goal is clear. Despite that the mapping can not map to the perfect MCS directly due to the uncertainty from the previous analysis; still, it is good enough.

doi:10.6342/NTU201900453

CHAPTER 6

CONCLUSION AND FUTURE WORK

We have investigated the design of reinforcement learning based OLLA mechanism in order to be more robust to the various communication environment and improve the well-known convergence issue in OLLA. The impact of the design of the reinforcement learning and the communication system are investigated when applying OLLA. The suggestions and results are listed in the following.

Firstly, PMI of user and paring users, CQI, and historical data of MCS, which are received ACK/NACK, are effective features. The proposed model is able to find out the relationship between these features and improve the performance.

Secondly, we verify the design of the training model based on the domain knowledge of the communication. It is noticing that with the proposed OLLA mechanism, the convergence steps can be improved by 38% with SU-MIMO feedback in comparison with the baseline method. Also, it is more robust to the different types of feedback. Furthermore, the throughput is increased by 14% in MU-MIMO and by 7% in NOMA+MUMIMO in comparison with the baseline method.

Thirdly, utilizing the potential of the capacity of single beam case and applying SU-MIMO are beneficial while applying OLLA. It can be seen that the throughput and fairness can be improved considerably with this setting.

Fourthly, we found that the convergence steps is not the only factor, which can affect the performance. The behavior of the chosen MCS have the influence as well. Thus, we design the reward shaping and control the number of the retransmissions to control the performance of the OLLA effectively. Moreover, controlling the number of the retransmissions have an extra benefit to prevent the transmission from suffering the overhead issue.

In short, the proposed OLLA can improve the performance significantly. The training procedure is effective for the convergence speed and the behavior of the process of converging. SU-MIMO feedback is suggested while operation OLLA. The constraint of the retransmissions provides the other possibility of the trained models.

doi:10.6342/NTU201900453

REFERENCES

[1] N. Wooseok, B. Dongwoon, L. Jungwon, and K. Inyup, "Advanced interference management for 5g cellular networks," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 52–60, 2014.

[2] T. Yoo, N. Jindal, and A. Goldsmith, "Multi-antenna downlink channels with limited feedback and user selection," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 7, 2007.

[3] M. Trivellato, F. Boccardi, and F. Tosato, "User selection schemes for mimo broadcast channels with limited feedback," in *Vehicular Technology Conference, 2007. VTC2007-Spring. IEEE 65th.* IEEE, 2007, pp. 2089–2093.

[4] J. Schaepperle and A. Regg, "Enhancement of throughput and fairness in 4g wireless access systems by non-orthogonal signaling," *Bell Labs Technical Journal*, vol. 13, no. 4, pp. 59–77, 2009.

[5] M.-J. Yang and H.-Y. Hsieh, "Moving towards non-orthogonal multiple access in next-generation wireless access networks," in *Communications (ICC), 2015 IEEE International Conference on.* IEEE, 2015, pp. 5633–5638.

[6] S. Timotheou and I. Krikidis, "Fairness for non-orthogonal multiple access in 5g systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1647–1651, 2015.

[7] H. Sun, Y. Xu, and R. Q. Hu, "A noma and mu-mimo supported cellular network with underlaid d2d communications," in *Vehicular Technology Conference (VTC Spring), 2016 IEEE 83rd.* IEEE, 2016, pp. 1–5.

[8] B. Kim, S. Lim, H. Kim, S. Suh, J. Kwun, S. Choi, C. Lee, S. Lee, and D. Hong, "Non-orthogonal multiple access in a downlink multiuser beamforming system," in *Military Communications Conference, MILCOM 2013-2013 IEEE.* IEEE, 2013, pp. 1278–1283.

[9] A. Sampath, P. S. Kumar, and J. M. Holtzman, "On setting reverse link target sir in a cdma system," in *Vehicular Technology Conference, 1997, IEEE 47th*, vol. 2. IEEE, 1997, pp. 929–933.

[10] M. G. Sarret, D. Catania, F. Frederiksen, A. F. Cattoni, G. Berardinelli, and P. Mogensen, "Dynamic outer loop link adaptation for the 5g centimeter-wave concept," in *European Wireless 2015; 21th European Wireless Conference; Proceedings of.* VDE, 2015, pp. 1–6.

[11] V. Buenestado, J. M. Ruiz-Aviles, M. Toril, S. Luna-Ramírez, and A. Mendo, "Analysis of throughput performance statistics for benchmarking lte networks," *IEEE Communications Letters*, vol. 18, no. 9, pp. 1607–1610, 2014.

doi:10.6342/NTU201900453

[12] F. Blanquez-Casado, G. Gomez, M. del Carmen Aguayo-Torres, and J. T. Entrambasaguas, "eolla: an enhanced outer loop link adaptation for cellular networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 20, 2016.

[13] S. Park, R. C. Daniels, and R. W. Heath, "Optimizing the target error rate for link adaptation," in *Global Communications Conference (GLOBECOM), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[14] R. A. Delgado, K. Lau, R. Middleton, R. S. Karlsson, T. Wigren, and Y. Sun, "Fast convergence outer loop link adaptation with infrequent updates in steady state," in *Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th*. IEEE, 2017, pp. 1–5.

[15] T. Ohseki and Y. Suegara, "Fast outer-loop link adaptation scheme realizing low-latency transmission in lte-advanced and future wireless networks," in *Radio and Wireless Symposium (RWS), 2016 IEEE*. IEEE, 2016, pp. 1–3.

[16] R. Daniels and R. W. Heath, "Online adaptive modulation and coding with support vector machines," in *Wireless Conference (EW), 2010 European*. IEEE, 2010, pp. 718–724.

[17] R. C. Daniels, C. M. Caramanis, and R. W. Heath, "Adaptation in convolutionally coded mimo-ofdm wireless systems through supervised learning and snr ordering," *IEEE Transactions on vehicular Technology*, vol. 59, no. 1, pp. 114–126, 2010.

[18] R. Bruno, A. Masaracchia, and A. Passarella, "Robust adaptive modulation and coding (amc) selection in lte systems using reinforcement learning," pp. 1–6, 2014.

[19] J. P. Leite, P. H. P. de Carvalho, and R. D. Vieira, "A flexible framework based on reinforcement learning for adaptive modulation and coding in ofdm wireless systems," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. IEEE, 2012, pp. 809–814.

[20] S. Yun and C. Caramanis, "Reinforcement learning for link adaptation in mimo-ofdm wireless systems," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*. IEEE, 2010, pp. 1–5.

[21] M. Rupp, S. Schwarz, and M. Taranetz, *The Vienna LTE-Advanced Simulators: Up and Downlink, Link and System Level Simulation*, 1st ed., ser. Signals and Communication Technology. Springer Singapore, 2016.

[22] X. Xia, S. Fang, G. Wu, and S. Li, "Joint user pairing and precoding in mu-mimo broadcast channel with limited feedback," *IEEE Communications Letters*, vol. 14, no. 11, pp. 1032–1034, 2010.

[23] L. Chen, Z. Chen, L. Liu, and B. Fu, "Successive precoding and user selection in mu-mimo broadcast channel with limited feedback," in *Wireless Telecommunications Symposium (WTS), 2014*. IEEE, 2014, pp. 1–5.

doi:10.6342/NTU201900453

[24] L. Yin, W. O. Popoola, X. Wu, and H. Haas, "Performance evaluation of non-orthogonal multiple access in visible light communication," *IEEE Transactions on Communications*, vol. 64, no. 12, pp. 5162–5175, 2016.

[25] Z. Shen, R. Chen, J. G. Andrews, R. W. Heath, and B. L. Evans, "Low complexity user selection algorithms for multiuser mimo systems with block diagonalization," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3658–3663, 2006.

[26] S. Tomida and K. Higuchi, "Non-orthogonal access with sic in cellular downlink for user fairness enhancement," in *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on.* IEEE, 2011, pp. 1–6.

[27] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction.* MIT press Cambridge, 1998, vol. 1, no. 1.

[28] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, *et al.*, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[29] K. I. Pedersen, G. Monghal, I. Z. Kovacs, T. E. Kolding, A. Pokhariyal, F. Frederiksen, and P. Mogensen, "Frequency domain scheduling for ofdma with limited and noisy channel feedback," in *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th.* IEEE, 2007, pp. 1792–1796.

[30] A. Duran, M. Toril, F. Ruiz, and A. Mendo, "Self-optimization algorithm for outer loop link adaptation in lte," *IEEE Communications Letters*, vol. 19, no. 11, pp. 2005–2008, 2015.

[31] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. Online Available at: http://tensorflow.org/

[32] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[33] T.-Y. Ho, P.-M. Lam, and C.-S. Leung, "Parallelization of cellular neural networks on gpu," *Pattern Recognition*, vol. 41, no. 8, pp. 2684–2692, 2008.

[34] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: efficient inference engine on compressed deep neural network," in *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on.* IEEE, 2016, pp. 243–254.

doi:10.6342/NTU201900453

[35] M. Shafique, T. Theocharides, C.-S. Bouganis, M. A. Hanif, F. Khalid, R. Hafız, and S. Rehman, "An overview of next-generation architectures for machine learning: Roadmap, opportunities and challenges in the iot era," in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2018*. IEEE, 2018, pp. 827–832.

[36] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*. IEEE, 2017, pp. 1–8.

[37] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

doi:10.6342/NTU201900453