國立交通大學

網路工程研究所

碩士論文

適用於長程演進網路上行傳輸之預測式資源排程法

An Estimation Based Resource Allocation Algorithm for LTE Uplink Transmission

研 究 生：張家愷

指導教授：趙禧綠　教授

中華民國 101 年 7 月

適用於長程演進網路上行傳輸之預測式資源排程法

An Estimation Based Resource Allocation Algorithm for LTE Uplink Transmission

研究生：張家愷　　Student：Chia-Kai Chang

指導教授：趙禧綠　　Advisor：His-Lu Chao

國立交通大學
網路工程研究所
碩士論文

A Thesis
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in

Computer Science

July 2012

Hsinchu, Taiwan, Republic of China

中華民國 101 年 7 月

適用於長程演進網路上行傳輸之預測式資源排程演算法

學生：張家愷　　　　　　　　　　　　　　　　指導教授：趙禧綠博士

國立交通大學資訊學院網路工程研究所

摘要

長期演進技術(LTE)為第三代合作夥伴計劃(3GPP)制定之標準，並且被公認為是邁向 4G 網路時期的一項很有前景的科技。由於在下行鏈路中使用正交分頻多工存取(OFDMA)技術，以及在上行鏈路中使用單載波分頻多工存取(SC-FDMA)技術，因此 LTE 相較於 3G 網路可以在頻寬最高 20MHz 上面提供極高的傳輸速率。

為了增加基地台的整體傳輸速率與使用者多樣性增益之目的， LTE 在以 OFDMA 為技術的下行鏈路中是採納傳統的通道相依排程演算法(CDS)。CDS 演算法會優先將各個資源區塊(RB)各別的配置給在這個 RB 頻道品質較好的使用者，然而這個方法若同樣的套用在以 SC-FDMA 為技術的上行鏈路中，則可能會讓整體的傳輸表現變得不盡理想。會發生這個問題的主要原因是因為 SC-FDMA 技術比 OFDMA 技術在配置資源的時候多了兩個限制，第一個是使用者在配置資源時必須符合連續限制，第二個則是針對一個使用者在所有被配置的 RBs 都必須使用相同的調變技術(MCS)。此次論文將把目標鎖定在探討上行鏈路以 SC-FDMA 為技術的排程演算法。由於上行鏈路的資源配置最佳解已經被証明為 NP-hard 問題，因此我們將藉由提出新的概念以發展出一個啟發式演算法，並且嘗試去逼近問題的最佳解。最後將以模擬評估本論文提出之演算法，結果顯示此方法在系統總傳輸率相較於傳統以 CDS 為基礎的方法有明顯的改善。

An Estimation Based Resource Allocation Algorithm for LTE Uplink Transmission

Student: Chia-Kai Chang Advisor: Dr. Hsi-Lu Chao

Institute of Network Engineering College of Computer Science National Chiao Tung University

Abstract

Long Term Evolution (LTE) is a promising technology for 4G mobile networks standard by 3rd Generation Partnership Project (3GPP). Due to utilize OFDMA in downlink and SC-FDMA in uplink, the LTE system is expected to provide significantly high throughput with 20 MHz spectrum allocation compared to 3G mobile networks.

To increase the cell throughput and multi-user diversity gain, Channel Dependent Scheduling (CDS) is implemented for the OFDMA-based multi-user scenario to allocate Resource Blocks (RBs) to users experiencing better channel conditions. Nevertheless, CDS may not perform well in SC-FDMA due to its two inherent constraints－one is contiguous RB assignment and the other is robust Modulation and Coding Scheme (MCS). In this thesis, since the optimization problem of resource allocation in SC-FDMA is NP-hard, we hence propose an estimation-based heuristic algorithm, which will take the two inherent constraints of SC-FDMA into consideration. In the algorithm, each time it will try to allocate one RB to the User Equipment (UE), which can maximize the current upper bound estimation. We evaluate the proposed algorithm by conducting simulation. The simulation results show that our method can achieve significant performance improvement in system throughput.

誌謝

兩年的時間過去了,開始寫論文和準備口試的日子也到來了。在碩士的求學時光裡，很開心能夠克服種種的考驗、困難與挫折，並且不斷的努力鞭策自我，致使自己能夠順利的完成課業和論文，準備畢業的到來。

首先，很感謝家裡的支持。即使家愷在國中和高中曾經沉迷於遊戲世界而不斷的想要放棄學業，家庭總是不斷的在背後默默的體諒、鼓勵和從旁協助，因此也讓我可以將從前不快樂的各種事情拋諸腦後，努力的往上學習。很高興自己終於到了畢業的時間，終究沒有辜負了家裡的培養。

家愷也很感謝我的指導老師，趙禧綠教授。在碩零期間，老師告訴了我們實驗室的方向，並且讓我們可以開始研讀自己有興趣的論文以尋找未來的研究題目。碩一修習課業、確定題目，碩二的論文研究、會議投稿、計畫申請、專利撰寫，到論文的正式完成和口試準備，老師在各個方面總是不斷的從旁給予協助和鼓勵，並且告知家愷正確的方向。

另外，我也要感謝實驗室裡的學長姐、同學以及學弟妹們。在家愷對於研究的知識不足的時候，學長姐總是會很熱心為我解答，並且告知哪邊會有我需要的文件。同學們總是在家愷心煩意亂的時候,很有義氣的陪我一起出去玩樂解悶,並且幫忙想辦法解決。碩一暑假的墾丁旅行雖然很不巧的遇到颱風,但是能夠跟大家一起郊遊實在是很難忘的經驗。而每星期的籃球運動也讓大家除了可以多多活動筋骨之外，更可以凝聚實驗室的向心力。

最後我要感謝交通大學給了我們一切的資源。實驗室和會議室，讓家愷可以努力的作研究和報告。不論是研究相關或課外書籍，圖書館提供了我們龐大的書籍資料。也很謝謝體育室提供家愷設備新穎的健身房和游泳池，讓我在研究疲憊之於，能夠藉由運動抒發壓力。

畢業的時間到了，也到了面臨兵役和工作的時候了。在工作之後，家愷會將這兩年從交通大學所學到的知識貢獻予社會，期望能夠在這個競爭激烈的世界中有所成就。而家愷也會將工作後得到的資源給予學校，讓交通大學可以不斷的提升競爭力並且培養出更多的卓越學生。在這，我要謝謝所有的家人、老師、同學和學校，謝謝各位一路的栽培與相挺。

Contents

摘　　要………………………………………………………………………………….I

Abstract …………………………………………………………………………..II

誌　　謝 III

Contents…………………………………………………………………...……IV

Lists of Table V

Lists of Figure...VI

Chapter 1　Introduction 1

1.1　Resource Block and Scheduling Procedure 2
1.2　Problem Statement... 5
1.3　Organization 7

Chapter 2　Related Work 8

Chapter 3　Proposed Algorithm 13

3.1　System Model 13
3.2　Problem Formulation 13
3.3　Heuristic Method 18
3.3.1.　Brief Introduction 19
3.3.2.　Method Description 20
3.3.3.　Detailed Example 27
3.3.4.　Pseudo Code ant Time Complexity 33

Chapter 4　Simulation Results 41

4.1　System Throughput... 42
4.2　Starvation Ratio 45
4.3　Influence of Window Size 47
4.4　Problem of UBERA.. 49

Chapter 5　Conclusion 50

References... 51

Lists of Table

Table I. The definition of parameters in problem formulation 15
Table II. The mapping table from SNR threshold to MCS 17
Table III. Example – Initial SNR table 27
Table IV. Example – Build window SNR table 28
Table V. Example – Choose starting RB to UE 29
Table VI. Example – Adjust SNR table 30
Table VII Example – Adjust window SNR table 31
Table VIII Example – Choose next RB 31
Table IX Example – Choose available allocated UEs 32
Table X Example – Upper bound estimation 33
Table XI Example – Scheduling result 33
Table XII Pseudo code of UBERA 34
Table XIII Pseudo code of Adjust SNR table 37
Table XIV Pseudo code of Adjust window SNR table 38
Table XV Parameter settings of the LTE UL system 42

Lists of Figure

Figure 1. Time domain view of the LTE 2
Figure 2. Time and frequency domain – user scheduling 3
Figure 3. Constraint of LTE UL resource allocation 5
Figure 4. Effect of choosing first RB to UE 6
Figure 5. Matrix algorithm and Search-tree based algorithm 9
Figure 6. The resource allocation results of RME, FME and MAD^E 9
Figure 7. IRME and ITRME 11
Figure 8. An illustration of parameter definition 15
Figure 9. The flowchart of UBERA 20
Figure 10. System throughput with optimum solution be compared 43
Figure 11. System throughput without optimum solution be compared 44
Figure 12. System throughput with different fading channel 45
Figure 13. Starvation ratio vs. Number of UEs 46
Figure 14. System throughput with different window size 47
Figure 15. An example of influence on different window size 48

Chapter 1 Introduction

The Long Term Evolution (LTE), marketed as 4G LTE, is one of important standard for wireless communication. The standard is developed by the 3rd Generation Partnership Project (3GPP), which is specified in 3GPP's Release 8 documents frozen in December 2008 with minor enhancement specified in Release 9 documents frozen in December 2009. With the reduced latency (5ms for small packet and 100ms for device wake up), higher data rate (peak data rate 300 Mbps for downlink while 75 Mbps for uplink for system bandwidth 20MHz and UE Category 5), flexible spectral usage (1.4, 3, 5, 10, 15, 20MHz with TDD and FDD available), better performance with high mobility equipment (maximum to 350km/h) and spatial multiplexing supported (single layer for Uplink (UL) per User Equipment (UE), up to 4 layers for Downlink (DL) per UE, and MU-MIMO supports for UL and DL), LTE is well-prepared to meet user expectation in a 10-year perspective and beyond [1].

To achieve these objectives, Orthogonal Frequency Division Multiple Access (OFDMA) has been selected as the DL access scheme for LTE cellular systems. Rather than transmitting a high-rate stream of data with a single carrier, OFDM makes use of a large number of closely spaced orthogonal subcarriers that are transmitted in parallel. Each subcarrier is modulated with a conventional modulation scheme (such as QPSK, 16QAM, or 64QAM) at a low symbol rate [2]. However, OFDMA is not suitable for the UL access scheme due to high Peak-to-Average Power Ratio (PAPR) which will shorten battery lifetime at User Equipment (UE), and leads 3GPP to look for a different transmission scheme for the LTE UL. Therefore, another modulation scheme－Single Carrier Frequency Division Multiple Access (SC-FDMA), is adopted for LTE UL transmission. Comparing to OFDMA, SC-FDMA performs a Discrete Fourier Transform (DFT) prior to the conventional Inverse Fast Fourier Transform (IFFT) operations, which spreads the data symbols over all the subcarriers and produces a virtual single-carrier structure [3]. This modification not only reduces consumed power significantly, but also keeps the inherent advantages of OFDMA such as high spectral efficiency and robustness to multipath fading.

1.1 Resource Block and Scheduling Procedure

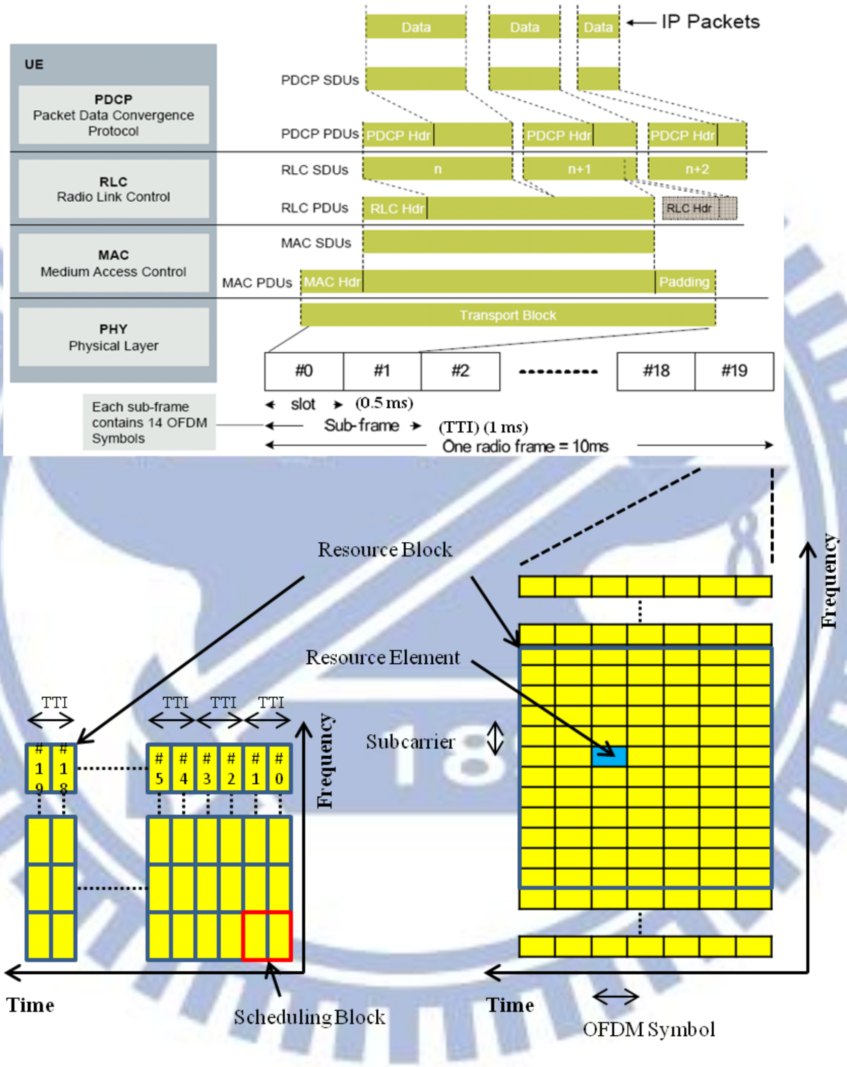


Figure 1. Time domain view of the LTE [4]

The time domain view and physical layer resource structure of LTE specified by 3GPP is shown in Fig. 1. As to time domain, a full frame is 10 ms but we normally think in terms of the 1 ms sub-frame, which is the entity that contains the Transport Block (TB). Within the TB (MAC PDU) are the MAC header, MAC SDUs and padding. Within the MAC SDU (RLC

PDU) are the RLC header and RLC SDUs, then within the RLC SDU (PDCP PDU) there can be a number of PDCP SDU IP Packets coming from network layer [4]. As to physical layer resource structure, the spectrum is divided into Resource Blocks (RBs), where a TB will be loaded into a number of RBs by scheduler at eNodeB. The number of RBs to load the specific TB is decided by the data size of TB and the data size can be loaded of single RB. RB is the basic LTE resource unit, which is a two dimensional rectangle wrapped by 12 adjacent spaced 15 kHz subcarriers in frequency domain and either 6 or 7 OFDM symbols in time domain depends on the Cyclic Prefix (CP). When a normal CP is used, the RB contains 7 symbols. When an extended CP is used, the RB contains 6 symbols. Two RBs consecutive in time domain form a single Scheduling Block (SB), which is the basic unit of bandwidth to be allocated to specific TB. The duration of SB equals the length of a sub-frame and represents a scheduling period, named Transmission Time Interval (TTI) [5]. In the latter of thesis, allocating single RB represents two time consecutive RBs of single SB. Besides, the smallest modulation structure in LTE is the Resource Element (RE). A RE is wrapped by one 15 kHz subcarrier and one OFDM symbol, which can carry several data bits depending on the adopted Modulation and Coding Scheme (MCS) mode. Since RB is composed of several REs as shown in Fig. 1, which is the smallest allocated unit, we define the sum data bits of a RB as “RB capacity”.

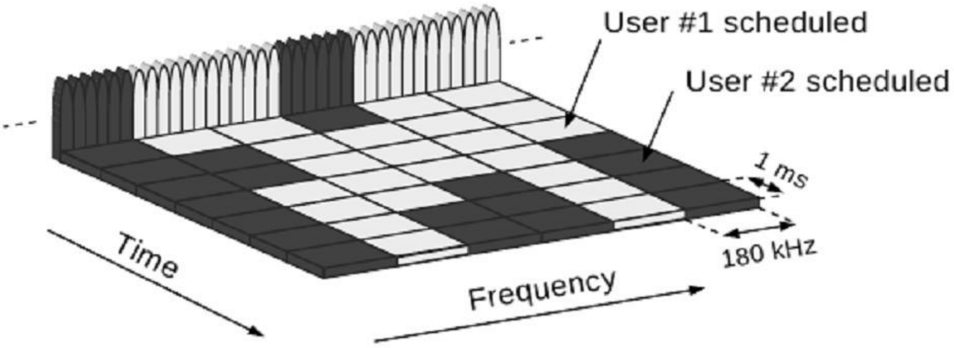


Figure 2. Time and frequency domain – user scheduling [6]

As active UE tries to request UL transmission resources from eNodeB, the UE uses Scheduling Request (SR) mechanism, which conveys a single bit of information indicating the UE has new data to transmit. The SR mechanism can be either Dedicated-SR (D-SR) or Random Access-based SR (RA-SR). While the UL of the UE is not time aligned or no Physical Uplink Control Channel (PUCCH) resources for D-SR are assigned to the UE, RA-SR must be used to (re-)establish time alignment. If the time is aligned and PUCCH resources are assigned, D-SR can be conveyed on PUCCH. After SR mechanism, portion of Physical Uplink Shared Channel (PUSCH) resources will be allocated to the UE. Then the Buffer Status Report (BSR) about the amount of data waiting in the UE is attached to the first UL transmission following the SR procedure on allocated PUSCH resources. After getting BSRs from UEs, the Packet Scheduler (PS) at eNodeB in each TTI makes decision on allocating RBs according to Sounding Reference Signals (SRSs) from all UEs. The SRS is a known sequence transmitted by UE periodically, where the UE-specific periodicity can be 2 / 5 / 10 / 20 / 40 / 80 / 160 / 320 ms as defined in [7] section 8.2. Getting SRS sequences from UEs, the SRSs are used at the eNodeB to extract the instantaneous Channel State Information (CSI) of the RBs of UEs, in which the CSI function is similar to the Channel Quality Indicator (CQI) in DL [8]. The better CSI of a RB refers to the higher RB capacity can achieve. Since the channel conditions are distinct among different RBs and uncorrelated for different UEs, PS will assign UE a portion of bandwidth that is in its favorite conditions. After PS decides resource allocation of UEs as illustrated in Fig. 2, eNodeB conveys UEs using Downlink Control Information (DCI) on Physical Downlink Control Channel (PDCCH). DCI contains information indicating MCS to be used and number of allocated RBs. Besides, DCI also tells the index of starting RB of UL resource allocation as well as number of contiguously allocated RBs.

1.2 Problem Statement

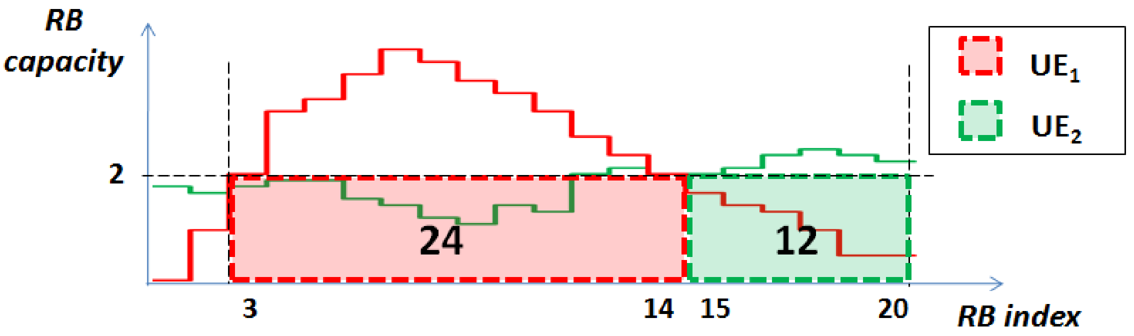


Figure 3. Constraint of LTE UL resource allocation

In an OFDMA-based multi-user system, RBs are allocated to the UEs that experience good channel conditions to maximize the multi-user diversity gain and increase the system throughput. As a result, CDS is well suitable for the LTE DL subsystem. However, for the LTE UL subsystem, RBs are allocated to a single UE in contiguous manner due to the inherent constraint of SC-FDMA [9]. This significantly reduces the freedom in resource allocation. In this thesis, we name this constraint "contiguous RB assignment". Another constraint which also affects the throughput performance of UL resource allocation is that a UE must adopt the same MCS for all allocated RBs [10]. Therefore, a UE can only utilize the lowest feasible RB capacity in its allocated RBs. We name this constraint "robust MCS mode". How the constraint of contiguous RB assignment and robust MCS mode affects the throughput performance is shown in Fig. 3. In Fig. 3, the *x* axis is RB index and *y* axis is RB capacity. The red and green curves are the envelope of the RB capacity for UE$_1$ and UE$_2$ for all observed RBs, respectively. For the constraint of contiguous RB assignment, we can see that the UEs must get RBs in the contiguous frequency band. On the other hand, for the constraint of robust MCS mode, we can see that even UE$_1$ has good RB capacity in the middle of its allocated RBs, these RBs' capacity must be slowed to the RB capacity of RB$_3$. Here, RB$_3$~RB$_{14}$ are allocated to UE$_1$ and

RB_{15}~RB_{20} are allocated to UE_2. RB_1 and RB_2 are unused. The system throughput performance of this example is 2*(20-3+1) = 36. However, it is easy to get better system throughput by allocating RB_4~RB_{12} to UE_1 and RB_{13}~RB_{20} to UE_2, where the system throughput promotes to 43.

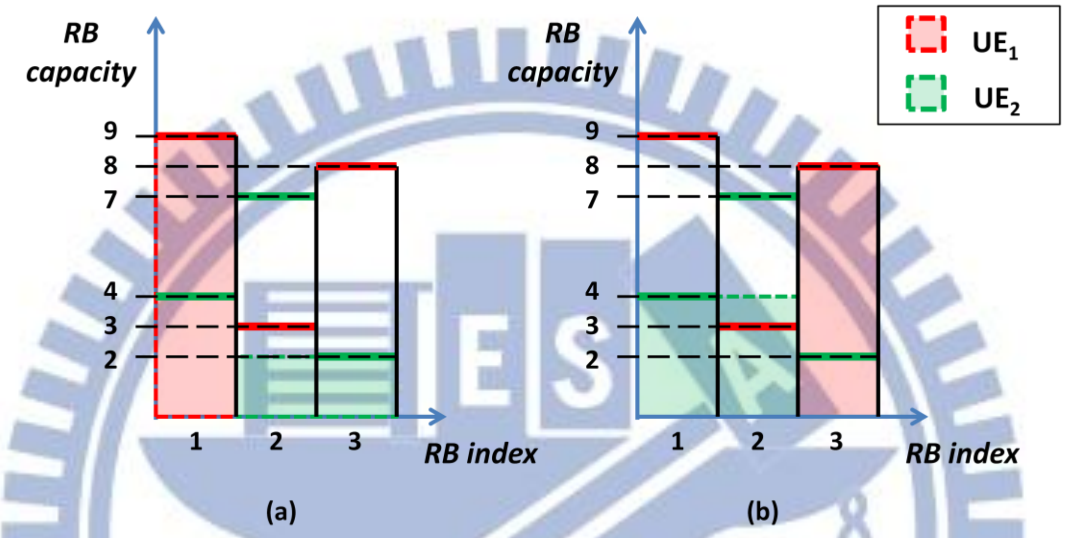


Figure 4. Effect of choosing first RB to UE

Besides, the method of choosing first RB to UE can also affect the system throughput. In Fig. 4(a), scheduler first finds the highest capacity and allocates RB_1 to UE_1. Since UE_1 doesn't have maximal capacity at adjacent RB_2, it then allocates to UE_2. Finally, RB_3 is also allocated to UE_2.The system throughput of Fig. 4(a) is 9+2+2=13. The scheduler of Fig. 4(b) is similar to Fig. 4(a) except that it first chooses the RB which has maximal capacity difference between best and second best capacity and allocates to the best UE. It first allocates RB_3 to UE_1. Then RB_1 and RB_2 are allocated to UE_2. The system throughput of Fig. 4(b) is 4+4+8=16. Thus it tells the CDS which greedily allocates RB to the best UE may not work well in SC-FDMA.

1.3 Organization

The remainder of the thesis is organized as follows. Related works are mentioned in chapter 2. The system model of LTE UL subsystem and the problem formulation are presented in chapter 3.1 and 3.2, respectively. The method description, detailed example, time complexity analysis and pseudo code of proposed estimation-based resource allocation algorithm are described in the remainder of chapter 3. The performance evaluation results are presented and discussed in chapter 4. Finally, the conclusions and future work are described in chapter 5.

Chapter 2 Related Work

There are many works focused on LTE DL resource allocation [11][12][13][14][15], and these proposed methods indeed have great improvement on system throughput performance. However, these methods cannot be applied to LTE UL resource allocation directly due to the constraint of “contiguous RB assignment.” Hence, there exists many works focused on improving system throughput of LTE UL. In this chapter, we first investigate methods of these LTE UL resource allocation works, and a summary is mentioned in the end of this chapter.

Calabrese et al. in [16] propose a search-tree based algorithm. At first, the authors divide the system bandwidth into Resource Chunks (RCs), which are equal sized and constituted by a set of consecutive RBs. The size of the RC is chosen to be a sub-multiple of the system bandwidth so that an integer number of UEs can be accommodated without creating bandwidth fragmentation. In algorithm description, this paper first described matrix algorithm, which continually allocates RC to UE with the highest metric. The author points the approach can provide significant gain over a random allocation, but does not achieve the global optimum. Thus, search-tree based algorithm is proposed. Rather than only considering the highest metric, the second highest metric is also considered to derive another sub-matrix. In this way, a binary search tree is derived where the best allocation corresponds to the path with highest sum of metrics. Fig. 5 illustrates the example of this algorithm with three UEs and three RCs. The left diagram shows the matrix algorithms while the right diagram shows the search-tree based algorithm. With more exhaustive search by proposed search-tree based algorithm, there exists more scheduling results. Thus, the scheduler can choose the best of scheduling results to globally improve the performance.

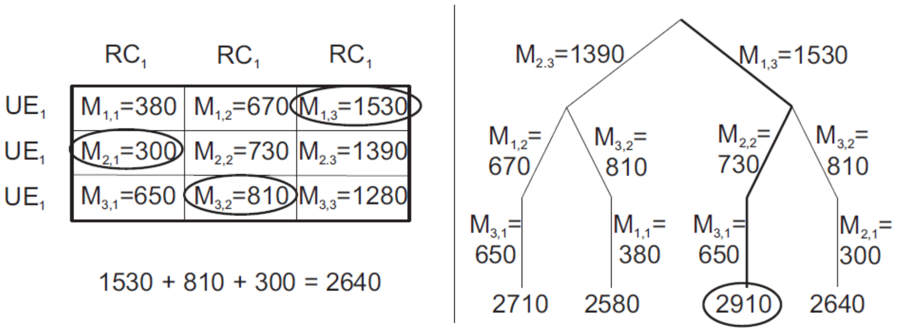


Figure 5. Matrix algorithm and Search-tree based algorithm [16]

Three channel aware scheduling algorithms, First Maximum Expansion (FME), Recursive Maximum Expansion (RME), and Minimum Area-Difference to the Envelope (MAD^E), are proposed in [17]. The main idea of FME is to assign RBs starting from the highest metric value and contiguously "expanding" the allocation on both sides. Each UE in FME is considered served whenever another UE having better metric is found. As to RME, the logic behind this algorithm is the same as FME, except that it performs a recursive search of the maximum. Finally, MAD^E is to derive the resource allocation that provides the minimum difference between its cumulative metric and the envelope-metric, i.e., the envelope of the users' metrics. Thus MAD^E can be seen as a generalized version of RME. These three proposed algorithms are show in Fig. 6. Each blank space between two lines in frequency means a RB while the curves indicate the corresponding RB capacity. The distinguished color blocks at the bottom of each diagram mean portion of RBs are allocated to different UEs.

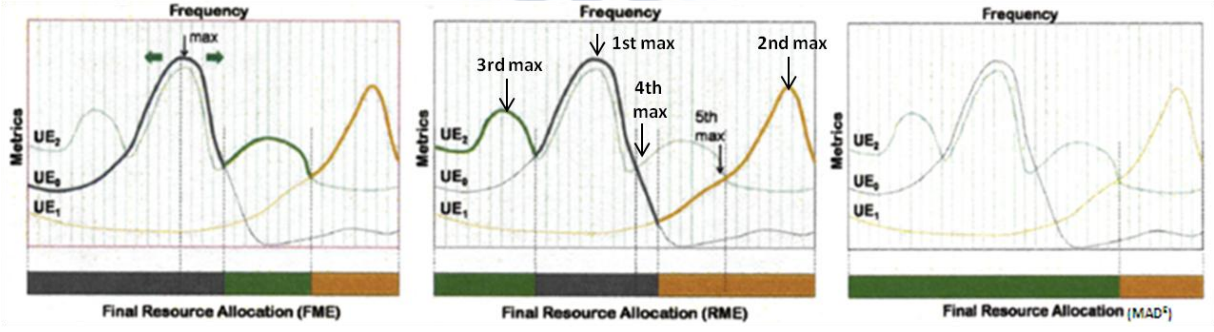


Figure 6. The resource allocation results of RME, FME and MAD^E [17]

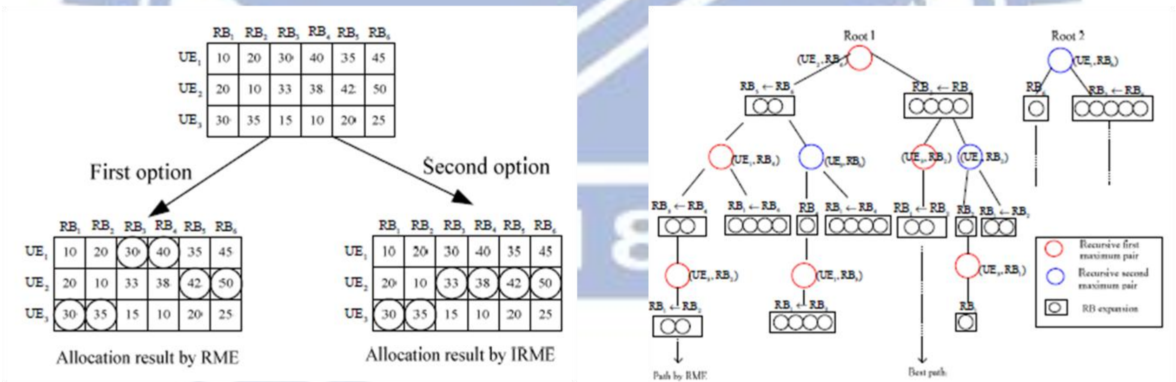
The authors in [18] adapt a different selection strategy based on difference between channel gains of users at a specific RB. By the limitation in allocating RBs to users, this paper find out some benefits and propose an algorithm of choosing an RB using difference in gain between RBs. The difference of a specific RB j can be defined as

$$\Delta_j = r_j\left(k_j^{(1)}\right) - r_j\left(k_j^{(2)}\right)$$

where $k_j^{(1)}$ is the best user, $r_j\left(k_j^{(1)}\right)$ is throughput of the best user, $k_j^{(2)}$ is the second best user and $r_j\left(k_j^{(2)}\right)$ is throughput of the second best user at RB j. In addition, a sub-algorithm is proposed for an already assigned user. When the selected RB is not adjacent to the already assigned RB or RBs of the same user, the scheduler decides whether the all RBs (from the selected RB to the already assigned RB or RBs) go to the user or abandon the selected RB. The scheduler hence calculates all the RB data rates of the user and all RB data rates of each available user at the selected RBs. Then, the scheduler compares data rates and decides whether to assign or not by the contiguity constraint. This proposed sub-algorithm reflects additional gain when a separated RB has a good channel condition.

In [19], two improved recursive maximum expansion scheduling algorithms for SC-FDMA are proposed. Compared with conventional recursive maximum expansion (RME) scheme in which UE can only expand the resource allocation on neighboring RBs with the highest metrics, in proposed improved recursive maximum expansion (IRME) scheme, higher degree of freedom in RB expansion is achieved by allowing RB expansion within certain ranking threshold *Tr*. The *Tr* means that there are *Tr* options for IRME to expand resource allocation on the neighboring RBs for UE. For option r, IRME will expand resource allocation on the neighboring RBs for UE only if its metric values are larger than or equal to the r th highest metric value, where $1 \leq r \leq Tr$. Then each option will output one allocation result.

Moreover, to further increase the flexibility in resource allocation, multiple surviving paths are introduced in proposed improved tree-based recursive maximum expansion (ITRME) scheme. Rather than considering only the pair (UE_m, RB_n) with the highest metric in the first step of RME and IRME, the UE_k with the second highest metric on the same RB_n is also considered in ITRME. For each sub-matrix ITRME consider again the best two UEs for the best RB. In this way this algorithm derives a binary search tree where the best allocation corresponds to the path with the highest sum of metrics. The two algorithms are illustrated in Fig. 7. By higher degree freedom in RB expansion, the IRME at left side of Fig. 7 gets more options. Moreover, ITRME at right side of Fig. 7 increases more allocation flexibility. Hence, it can get much more available options compared to RME and IRME. However, the more options mean the higher degree of time and space complexity consumed.



	RB_1	RB_2	RB_3	RB_4	RB_5	RB_6
UE_1	10	20	30	40	35	45
UE_2	20	10	33	38	42	50
UE_3	30	35	15	10	20	25

Figure 7. IRME and ITRME [19]

In [20], the optimization formulation of packet scheduling problem in LTE uplink is proposed in this paper. The optimization formulation defines the packet scheduling problem as a transform of the knapsack problem, in which the number of RBs allocated to each user corresponds to the weight of each item and evaluation metrics corresponds to the value of each item. Besides, the author also utilizes the Integer Linear Programming (ILP) method to provide the feasible optimum resource allocation based on the combination of allocable resources with various constraints in LTE uplink. Moreover, to reduce the complexity of the ILP, a limitation on the valid combination of allocable resources is also considered.

All the related works described above have indeed taken the contiguous RB assignment constraint into consideration while allocating LTE UL RBs. However, a common issue of these works is that each UE is allowed to using different MCS modes on its allocated RBs. This is an improperly assumption since the modulation function in physical layer can select only one MCS mode to modulate TB [21], which is decided at MAC layer and conveyed to UE through DCI. In other words, each UE can only use the robust MCS mode on its allocated RBs. Thus, recently, the author in [22] proposes two heuristic algorithms – TTRA and STRA, which take the robust MCS mode constraint into consideration. However, these two algorithms are still based on RME to modify, which doesn’t consider robust MCS mode. Therefore, the improvement on these two algorithms is limited by inherent problem of RME. Thus, in this thesis, we propose a novel algorithm by taking these two constraints into consideration, which doesn’t be limited in CDS or RME algorithms.

Chapter 3 Proposed Algorithm

In this chapter, we first describe the scenario of LTE UL system. Followed, the scheduling problem is formulated as Integer Linear Programming (ILP). Finally, the heuristic algorithm is proposed to solve this problem.

3.1 System Model

In this thesis, we consider a cellular network which consists of a fixed serving eNodeB and n active UEs. The UL bandwidth of this cellular network is divided into m RBs. Due to the inherent constraint of SC-FDMA, in each scheduling period (or TTI), RBs assigned to a single UE must be in contiguous manner. Besides, since MIMO is not the focal point in this research, each RB can be only assigned to at most one UE. Owing to the constraint of robust MCS mode, a UE operates at the same MCS mode in its' all assigned RBs. Since channel conditions typically depend on channel frequencies, user locations, and time slots, each RB has user-dependent and time-varying channel conditions. These conditions are transmitted from active UEs to eNodeB periodically. Thus, eNodeB can know all the active UEs' channel conditions.

3.2 Problem Formulation

By expressing as ILP problem, the objective of our proposed algorithm is to maximize the LTE UL system throughput, while keeping the constraint at the same time. Before starting formulate problem, we first introduce some parameters listed in Table I. We first define $S_i(t)$ as the set of assigned RBs of UE$_i$ and $|S_i(t)|$ represents the number of RBs in set $S_i(t)$. Also, let $\delta_{i,j}(t)$ and $r_i(S_i(t))$ be the channel signal-to-noise ratio (SNR) measured by UE$_i$ at RB$_j$ and the robust RB capacity at RBs that UE$_i$ operate, respectively. Besides, $x_{i,j}(t)$ is an allocation indicator of $\mathrm{UE_i}$ at $\mathrm{RB_j}$. If $\mathrm{RB_j}$ is allocated to $\mathrm{UE_i}$ by scheduler, $x_{i,j}(t)$ is assigned to 1. Otherwise, $x_{i,j}(t)$ is assigned to 0. An illustration of these parameters is shown in Fig.8. In this example, $S_i(t)$ can be expressed as

$$S_i(t) = \{\, \alpha, \alpha+1, \alpha+2, \dots, \beta-1, \beta \,\}$$

Hence, $|S_i(t)|$ which counts the number in $S_i(t)$ can be derived as

$$|S_i(t)| = \beta - \alpha + 1$$

From $S_i(t)$, we know the scheduler allocates RBs $\mathrm{RB}_\alpha, \mathrm{RB}_{\alpha+1}, \dots, \mathrm{RB}_\beta$ to $\mathrm{UE_i}$. Consequently, the allocation indicator $x_{i,j}(t)$ of $\mathrm{UE_i}$ at all the RBs can be assigned as follows:

$$x_{i,\alpha}(t) = x_{i,\alpha+1}(t) = \cdots = x_{i,\beta}(t) = 1$$

$$x_{i,1}(t) = x_{i,2}(t) = \cdots = x_{i,\alpha-1}(t) = 0$$

$$x_{i,\beta+1}(t) = x_{i,\beta+2}(t) = \cdots = x_{i,\mathrm{m}}(t) = 0$$

The particular $f(\delta_{i,j}(t))$ is the rectangle height of $\mathrm{UE_i}$ at $\mathrm{RB_j}$, in which the *f*() is an mapping function which can get the maximum available data rate of a specific $\delta_{i,j}(t)$ through looking up the MCS table. Thus, $f(\delta_{i,\alpha}(t))$ and $f(\delta_{i,\beta}(t))$ are the height of $\mathrm{UE_i}$ at RB_α and RB_β, respectively. Finally, having above information, we can get $r_i\big(S_i(t)\big)$ as

$$r_i\big(S_i(t)\big) = f\big(\min\{\delta_{i,\alpha}(t), \delta_{i,\alpha+1}(t), \delta_{i,\alpha+2}(t), \dots, \delta_{i,\beta-1}(t), \delta_{i,\beta}(t)\}\big).$$

Table I. The definition of parameters in problem formulation

Parameter	**Definition**
$S_i(t)$	The set of assigned RBs of UE_i
$\lvert S_i(t)\rvert$	The number of RBs in set $S_i(t)$
$r_i\big(S_i(t)\big)$	The robust RB capacity in $S_i(t)$
$\delta_{i,j}(t)$	The SNR value of UE_i at RB_j
$x_{i,j}(t)$	The allocation indicator of UE_i at RB_j

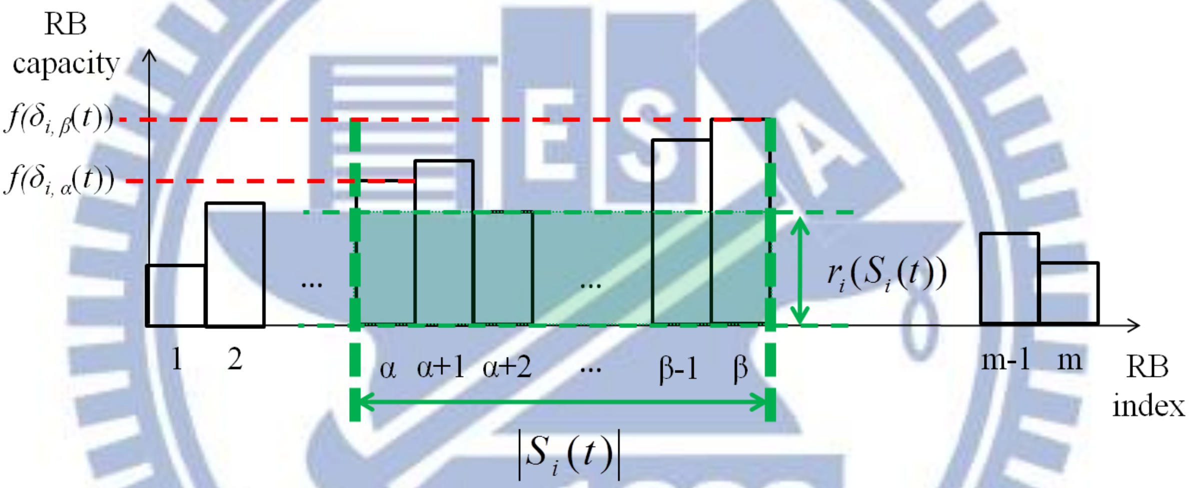


Figure 8. An illustration of parameter definition

Regarding contiguous RB assignment, this constraint can be expressed as follows:

$$x_{i,j}(t) = 0, \text{for all } j \geq\ n + 2 \qquad (1)$$

$$\text{When } x_{i,n}(t) = 1, x_{i,n+1}(t) = 0$$

The equation (1) descripts that as RB_n is allocated to UE_i while RB_{n+1} is not, the RBs from $\mathrm{RB}_{n+2}, \mathrm{RB}_{n+3}$ to RB_m cannot be able to UE_i anymore. Thus, this equation ensures the constraint of contiguous RB assignment.

In spite of the contiguous RB assignment constraint, the other inherent constraint, robust MCS mode, should also be followed. Since UE has different channel quality $\delta_{i,j}(t)$ among RBs, the maximum available data rate and corresponding MCS can hence be different. For this reason, the height $r_i(S_i(t))$ is introduced, which represents the least RB capacity among all allocated RBs of UE$_i$. As the specification defined, the active UEs report the instantaneous UL channel quality, as known as SNR value $\delta_{i,j}(t)$, through SRS coding sequences on each RB$_j$ to the serving eNodeB, periodically. The eNodeB maps the SNR value to achievable MCS mode on all RE within each RB, and then RB capacity of each RB is determined. The approach is proposed in [23]. The height $r(S_i(t))$ of a RB rectangle can be expressed as follows:

$$r_i(S_i(t)) = f\left(\min_{j \in S_i(t)} \delta_{i,j}\right) \tag{2}$$

where *f()* function as mentioned before is to derive the maximum available RB capacity from specific channel SNR value $\delta_{i,j}(t)$. In general, *f* function can be implemented by a look-up table in eNodeB as shown in Table II. The RB capacity can be calculated by equation (3) [24].

$$f(\cdot) = RB\ capacity = \frac{nbits}{symbol} \times \frac{nsymbols}{slot} \times \frac{nslots}{TTI} \times \frac{nsc}{RB} \tag{3}$$

where *nbits/symbol* computes how many bits carried per symbol, as the same as per RE, and it can be retrieved directly by which MCS mode is employed. *nsymbols/slot* is the number of symbols per slot, *nslot/TTI* is the number of slots per TTI and *nsc/RB* is the number of sub-carriers per RB. Table II shows the mapping table in which the MCS mode begins from QPSK (1/2) to 64QAM (3/4). In this table, $\delta_{i,j}(t)$ must be larger than or equal to the SNR threshold, so the corresponding MCS mode can be applied. For example, one UE$_i$ at RB$_j$

having $\delta_{i,j}(t)$ =8 means this RB can be modulated as QPSK (1/2), QPSK (2/3), QPSK (3/4) or 16QAM (1/2), but cannot be modulated as 16QAM (2/3) and the rest MCS.

Table II. The mapping table from SNR threshold to MCS

MCS mode	SNR threshold (dB)	nbits/RE	RB capacity (Kbps)
QPSK(1/2)	1.7	1	168
QPSK(2/3)	3.7	1.33	224
QPSK(3/4)	4.5	1.5	252
16QAM(1/2)	7.2	2	336
16QAM(2/3)	9.5	2.66	448
16QAM(3/4)	10.7	3	504
64QAM(2/3)	14.8	4	672
64QAM(3/4)	16.1	4.5	756

In addition to the constraints of contiguous RB assignment and robust MCS mode formulated at equation (1), (2) and (3). There still exist some inherent constraints because of our system model assumption. Equation (4) is expressed as follows:

$$\sum_{i=1}^{n} x_{i,j}(t) \leq 1, \quad \text{for all } j \tag{4}$$

Since the MIMO is not taken into consideration, this equation illustrates that each RB$_j$ can be assigned to at most one UE. Moreover, $|S_i(t)|$ of each UE$_i$ is counted through:

$$|S_i(t)| = \sum_{j=1}^{m} x_{i,j}(t), \quad \text{for all } i \tag{5}$$

To ensure the total allocated RBs at a specific TTI t doesn't exceed the supported UL bandwidth, we used the following equation to limit the number of allocated RBs:

$$\sum_{i=1}^{n} |S_i(t)| \leq m, \quad \forall\, S_i(t) \neq \emptyset \qquad (6)$$

Finally, we assume the scheduler performs resource allocation per TTI t. Therefore, the system throughput maximization problem can be well-formulated by equation (7):

$$max \sum_t \sum_i |S_i(t)|\, r\big(S_i(t)\big) \qquad (7)$$

Subject to the constraints of (1)(2)(3)(4)(5)(6)

3.3 Heuristic Method

Since the problem of LTE UL scheduling described as above has been proven to be NP-hard [25]. Thus it is not possible to achieve the maximal system throughput performance in polynomial time. To compromise with the complexity, we present a heuristic algorithm: Upper Bound Estimation Resource Allocation (UBERA).

Before starting, some additional parameters are introduced to help describe our algorithm. As to the system model described, we assume there are m RBs and n UEs in the cellular network at TTI t. Let $\mathcal{M}$ and $\mathcal{M}'$ be the sets of non-checked and checked RBs, respectively. Also, let $\mathcal{N}$ and $\mathcal{N}'$ be the sets of available scheduled and non-available scheduled UEs, respectively. Initially $\mathcal{M}$={RB$_1$, RB$_2$, …, RB$_m$}, and $\mathcal{N}$={UE$_1$, UE$_2$, …, UE$_n$,}; $\mathcal{M}'$ and $\mathcal{N}'$ are both $\emptyset$. For simplicity, we use |●| to indicate the number of elements in a set. For example, initially $|\mathcal{M}| = m$ and $|\mathcal{N}| = n$. Besides, $\delta_{i,j}$, where $i \in \mathcal{N}$, $j \in \mathcal{M}$, represents the measured SNR value of UE$_i$ in RB$_j$.

3.3.1. Brief Introduction

The main characteristic of UBERA is that as each time it tries to allocate one RB to UE, the algorithm gets all available allocated UEs on the RB, and calculates the upper bound after allocating to the UEs, respectively. After getting all upper bound values of different UEs, UBERA gives the RB to the UE which has the maximal upper bound. The reason of adopting this method is that the common existing scheduling algorithms for LTE UL allocate the RB to the UE, which has the highest SNR at the RB or can derive the current maximal sum system throughput, but doesn’t consider the future influence after this allocation. Thus, it is possible that the allocation can temporarily increase system throughout, but may become worse in the latter scheduling due to robust MCS mode.

The flowchart of UBERA is illustrated in Fig.9. Firstly, the algorithm calls “Build window SNR table.” In this procedure, we take SNR table, which records all the active UEs to RBs channel conditions, as reference to build a brand-new table called window SNR table. After the table is built, this goes to “Choose starting RB to UE” procedure, which decides the first RB to be checked. Besides, the checked RB is also allocated to one active UE here. To keep the constraint of robust MCS mode, we hence call “Adjust SNR table” which adjusts the values recorded in SNR table. Since the window SNR table is built based on SNR table, the “Adjust window SNR table” is called to adjust the values recorded in window SNR table. Before start allocating next RB to UE, we check whether there are available RBs unchecked. If all the available RBs in system are checked, the algorithm is terminated here. In case that there is RB unchecked, the algorithm in “Choose nest RB” procedure chooses one RB from unchecked RBs. Since the later “Upper bound estimation” is the core of this algorithm, which consumes most of the time, taking all the UEs to this procedure to implement estimation will waste too much time. Thus, we call “Choose available allocated UEs” to choose some candidate UEs. In “Upper bound estimation,” the upper bounds of candidate UEs are calculated and the chosen RB is allocated to the candidate UE whom has the maximal upper bound. The “Adjust SNR table” and “Adjust window SNR table” are called to follow robust MCS mode. Finally, this algorithm backs to check whether there RBs unchecked.

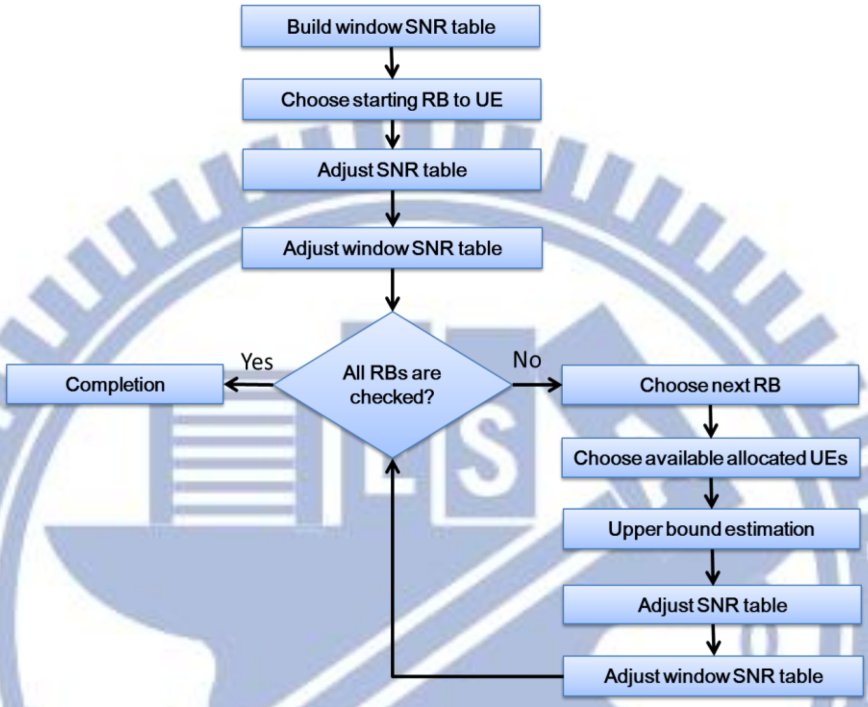


Figure 9. The flowchart of UBERA

3.3.2. Method Description

In this paragraph, we descript the procedures of this algorithm in detail one by one. The procedures are described follows.

(1) Build window SNR table

Before start scheduling, the algorithm first defines a term called *window*. Here, we denote it as *w*. Besides, the *w* has a size called *window size*, which is an odd number less than or equal to *m* and denotes it as *ws*. The *w* is a rectangle that can once cover several RBs based on *ws* for a single UE at a RB. At this procedure, we first place the *w* on RB_1 of UE_1. Second, the minimum SNR in the *w* is selected as *window SNR*, and put the value on the RB_1 of UE_1 in *window table*. Here, the *window SNR* of UE_i at RB_j as $wSNR_{i,j}$ is expressed as:

$$wSNR_{i,j} = \min\left(\delta_{i,j-\frac{ws-1}{2}}, \delta_{i,j-\frac{ws-1}{2}+1}, \dots, \delta_{i,j-\frac{ws-1}{2}+ws-1}\right)$$

We then slide the *window* to RB_2 of UE_1, and get $wSNR_{1,2}$. After the *w* has put on every RBs of each UE, we finally completely build *window table* with values from $wSNR_{1,1}$, $wSNR_{1,2}$, $wSNR_{1,3}$,…, $wSNR_{1,m}$, $wSNR_{2,1}$,… to $wSNR_{n,m}$. The reason of building the *window table* is that the LTE UL must keep to contiguous RB assignment and robust MCS mode. Hence, the minimum SNR in *w* can be taken as the possible average SNR after allocating the RB, which we predict in prior.

(2) Choose starting RB to UE

For each RB_j, it first calculates the difference between maximal and second maximal $wSNR_{i,j}$, which is denoted as Δ_j.The Δ_j of a specific RB_j is expressed as:

$$\Delta_j = \max{}^1\left(wSNR_{i,j}\right) - \max{}^2\left(wSNR_{i,j}\right)$$

where $\max^1\left(wSNR_{i,j}\right)$ and $\max^2\left(wSNR_{i,j}\right)$ is the maximal and second maximal $wSNR_{i,j}$ at RB_j. It then picks RB_j, which has the maximal Δ_j, and allocates to UE whom owns $\max^1\left(wSNR_{i,j}\right)$ at RB_j. However, if there are more than one Δ_j having same maximal difference, the scheduler then picks the RB_j, which has the maximal $\max^1\left(wSNR_{i,j}\right)$ among these same maximal Δ_j, and allocates to the UE whom owns this $\max^1\left(wSNR_{i,j}\right)$. Then, the procedure removes the just allocated RB_j from $\mathcal{M}$ to $\mathcal{M}'$.

(3) Adjust SNR table

Since the robust MCS mode needs to be guaranteed, it tries to adjust all $\delta_{i,j}$ of a specific

UE$_i$. After getting one $\delta_{i,j}$ from previous procedure (“Choose starting RB to UE” or “Upper bound estimation”), where RB$_j$ is just allocated to UE$_i$, the sub-program checks all the $\delta_{i,j}$ of UE$_i$, which denotes as $\delta_{i,j'}$. If $\delta_{i,j'}$ is larger than $\delta_{i,j}$ and RB$_j$ not equals to RB$_{j'}$, the $\delta_{i,j'}$ is revised to $\delta_{i,j}$.

(4) Adjust window SNR table

In this sub-program, it revises the *window table*, which is first built at “Build window SNR table.” Here, we check the UEs in $\mathcal{N}$ one by one. At first, it puts *w* on the first left side non-allocated RB of allocated RBs of UE in $\mathcal{N}$. Then, if the most left side allocated RB is not the current UE getting from $\mathcal{N}$, the $wSNR_{i,j}$ is revised as

$$wSNR_{i,j} = \min\,(\delta_{i,j-\frac{ws-1}{2}}, \delta_{i,j-\frac{ws-1}{2}+1}, \ldots, \delta_{i,j}.)$$

We then slide *w* to the second left side non-allocated RB, and revise the $wSNR_{i,j}$ as

$$wSNR_{i,j} = \min\,(\delta_{i,j-\frac{ws-1}{2}}, \delta_{i,j-\frac{ws-1}{2}+1}, \ldots, \delta_{i,j+1}).$$

The *w* continually slides left and revises $wSNR_{i,j}$. Until the $wSNR_{i,j}$ is revised as

$$wSNR_{i,j} = \min\,(\delta_{i,j-\frac{ws-1}{2}}, \delta_{i,j-\frac{ws-1}{2}+1}, \ldots, \delta_{i,j-\frac{ws-1}{2}+ws-2})$$

the UE stops. However, if the UE in $\mathcal{N}$ equals to the UE of most left side allocated RB, it first towards right looking for the most right allocated RB of this UE, and denotes the RB as RB$_{j''}$. Then, the *w* is similarly puts on the first left side non-allocated RB, RB$_j$. If $j'' \geq j - \frac{ws-1}{2} + ws - 1$, it stops, else the $wSNR_{i,j}$ is revised as

$$wSNR_{i,j} = \min\left(\delta_{i,j-\frac{ws-1}{2}}, \delta_{i,j-\frac{ws-1}{2}+1}, \dots, \delta_{i,j''}\right)$$

Then we slide *w* left, and revise the $wSNR_{i,j}$ as

$$wSNR_{i,j} = \min\left(\delta_{i,j-\frac{ws-1}{2}}, \delta_{i,j-\frac{ws-1}{2}+1}, \dots, \delta_{i,j''}\right)$$

until $j'' = j - \frac{ws-1}{2} + ws - 1$, it stops. After the $wSNR_{i,j}$ at left side non-allocated RBs of UEs in $\mathcal{N}$ is totally adjusted, the procedure tries to revise the $wSNR_{i,j}$ at right side non-allocated RBs of allocated RBs, which is similar to the description described above. At first, it puts *w* on the first right side non-allocated RB of allocated RBs of UE in $\mathcal{N}$. If the most right side allocated RB is not the current UE getting from $\mathcal{N}$, the $wSNR_{i,j}$ is revised as

$$wSNR_{i,j} = \min\left(\delta_{i,j}, \delta_{i,j+1}, \dots, \delta_{i,j-\frac{ws-1}{2}+ws-1}\right)$$

We then slide *w* to the second right side non-allocated RB, and revise the $wSNR_{i,j}$ as

$$wSNR_{i,j} = \min\left(\delta_{i,j-1}, \delta_{i,j}, \dots, \delta_{i,j-\frac{ws-1}{2}+ws-1}\right)$$

The *w* continually slides right and revises $wSNR_{i,j}$. Until the $wSNR_{i,j}$ is revised as

$$wSNR_{i,j} = \min\left(\delta_{i,j-\frac{ws-1}{2}+1}, \delta_{i,j-\frac{ws-1}{2}+2}, \dots, \delta_{i,j-\frac{ws-1}{2}+ws-1}\right)$$

the UE stops. However, if the UE in N equals to the UE of most right side allocated RB, it first towards left looking for the most left allocated RB of this UE, and denotes the RB as RB$_{j''}$. Then, the *w* is similarly puts on the first right side non-allocated RB, RB$_{j}$. If $j'' \leq j - \frac{ws-1}{2}$, it stops, else the $wSNR_{i,j}$ is revised as

$$wSNR_{i,j} = \min(\delta_{i,j''}, \delta_{i,j''+1}, \dots, \delta_{i,j-\frac{ws-1}{2}+ws-1})$$

Then we slide *w* right, and revise the $wSNR_{i,j}$ as

$$wSNR_{i,j} = \min(\delta_{i,j''}, \delta_{i,j''+1}, \dots, \delta_{i,j-\frac{ws-1}{2}+ws-1})$$

until $j'' = j - \frac{ws-1}{2}$, it stops.

(5) Choose next RB

At this procedure, the scheduler tries to determine the next RB from $\mathcal{M}$ to allocate. Let the first un-allocated RB on the left side of allocated RBs be RB$_{left}$, and the first un-allocated RB on the right side of allocated RBs be RB$_{right}$. The decision value of RB$_{left}$ is $wSNR_{i',left}$, where RB$_{left+1}$ allocates to UE$_{i'}$. Besides, the decision value of RB$_{right}$ is $wSNR_{i'',right}$, where RB$_{right-1}$ allocates to UE$_{i''}$. However, if RB$_{left+1}$ allocates to virtual UE (the term here means not allocated to any real UE), the decision value of RB$_{left}$ will be the maximal $wSNR_{i,left}$ among all the UE$_{i}$s, where $i \epsilon \mathcal{N}$ and available to be scheduled at RB$_{left}$. Similarly, if RB$_{right-1}$ allocates to virtual UE, the decision value of RB$_{right}$ will be the maximal $wSNR_{i,right}$ among all the UE$_{i}$s, where $i \epsilon \mathcal{N}$ and available to be scheduled at RB$_{right}$. The scheduler then picks RB$_{right}$ as next RB to allocate if the decision value of RB$_{right}$ is larger than or equal to the decision value of RB$_{left}$, else picks RB$_{left}$. However, if there is no RB$_{left}$, it returns RB$_{right}$ directly, and returns RB$_{right}$ if there is no RB$_{left}$. Finally, if there is no remaining RB to schedule, the algorithm terminates at here.

(6) Choose available allocated UEs

After picking RB (denotes as RB_{pick}, where $\text{RB}_{pick} \epsilon \mathcal{M}$) from “Choose next RB”, the scheduler determines the available UEs at RB_{pick} here. If the side of RB_{pick}, which is allocated before (denotes as $\text{RB}_{scheduled}$, where $\text{RB}_{scheduled} \epsilon \mathcal{M}'$), is to virtual UE, it goes to “Upper bound estimation” directly (1). However, If $\text{RB}_{scheduled}$ is not to virtual UE and there are some UEs’ $wSNR_{i,pick}$, where $i \epsilon \mathcal{N}$ and available to be scheduled at RB_{pick} (denotes as $\text{UEs}_{available}$), is larger than or equal to $wSNR_{i',pick}$, where $\text{RB}_{scheduled}$ allocates to $\text{UE}_{i'}$, it then picks all these larger valued UEs with $\text{UE}_{i'}$ to “Upper bound estimation” (2). On the contrary, if $wSNR_{i,pick}$ of all $\text{UEs}_{available}$ is less than $wSNR_{i',pick}$, it checks as follows. First, it gets all the UEs of $\text{UEs}_{available}$, which has the maximal $wSNR_{i,j}$ at RB_{pick} among all the RBs of the UE, where RBs $\epsilon \mathcal{M}$ and calculates the gain Δ_{gain} as follows:

$$\Delta_{gain} = \left(wSNR_{i',pick} - wSNR_{i,pick}\right) - \left(wSNR_{i,pick} - wSNR_{i,j'}\right),$$

where $wSNR_{i,j'}$ is the second maximal among all the RBs $\epsilon \mathcal{M}$ of the UE. Here, the first term of the formula means the gain after allocating RB_{pick} to $\text{UE}_{i'}$, and the second term means the gain after allocating $\text{RB}_{j'}$ to UE_{i}, respectively. If there exists UEs whose $\Delta_{gain} \leq 0$, which means allocating RB_{pick} to $\text{UE}_{i'}$ may cause the gain of the UEs down, then the scheduler picks all UEs whose $\Delta_{gain} \leq 0$ with $\text{UE}_{i'}$ to “Upper bound estimation” (3). However, as all the UEs’ $\Delta_{gain} > 0$ or no $\text{UEs}_{available}$ have maximal $wSNR_{i,j}$ at RB_{pick}, and this allocation can promote the system throughput, it allocates RB_{pick} to $\text{UE}_{i'}$, removes RB_{pick} from $\mathcal{M}$ to $\mathcal{M}'$; else it goes to “Upper bound estimation” (4).

(7) Upper bound estimation

At this procedure, the scheduler calculates the upper bound after allocating RB_{pick} to available UEs, which determines at “Choose available allocated UEs”, respectively, and allocates to the UE whom has maximal upper bound. In (1) and (4) of “Choose available allocated UEs”, the available UEs are $\mathrm{UEs}_{available}$. The available UEs of (2) and (3) are picked at “Choose available allocated UEs”. Here, let the UEs available to be pseudo-allocated at left side of allocated RBs be UEs_{left}, which belongs to $\mathcal{N}$ and not includes the UE, whom obtains the most right side allocated RBs, if the UE is blocked at left side by another UE. Besides, let the UEs available to be pseudo-allocated at right side of allocated RBs be UEs_{right}, which belongs to $\mathcal{N}$ and not includes the UE, whom obtains the most left side allocated RBs, if the UE is blocked at right side by another UE. The upper bound of a specific UE_i is predicted as follows.

At first, the scheduler pseudo-allocates RB_{pick} to UE_i, and calculates the temporal upper bound. If UE_i not equals to $\mathrm{UE}_{i'}$, whom obtains $\mathrm{RB}_{scheduled}$, the $\mathrm{UE}_{i'}$ thus cannot be used at current prediction, since it has been blocked by UE_i. The scheduler here kicks $\mathrm{UE}_{i'}$ from UEs_{left}, if RB_{pick} is at left side of allocated RBs; else kicks $\mathrm{UE}_{i'}$ from UEs_{right}. Second, it sorts RBs $\epsilon\mathcal{M}$, which doesn’t include RB_{pick} based on maximal $\delta_{i,j}$, where $i\epsilon\mathrm{UEs}_{left}$ for left side non-allocated RBs or $i\epsilon\mathrm{UEs}_{right}$ for right side non-allocated RBs. After sorting, it picks the highest priority RB and calculates the temporal upper bound after pseudo-allocating to UEs_{left} or UEs_{right} depends on the place of the RB, respectively. Since all the temporal upper bounds of pseudo-allocating the highest priority RB to different UEs_{left} or UEs_{right} have been completed, the scheduler chooses the maximal bound of this pseudo-allocating. It then picks the second highest priority RB and does the same thing, and stops until all the RBs in sorting results have been picked. After all prediction is completed, each UE available to be allocated at RB_{pick} has upper bound estimation. The scheduler then real allocates RB_{pick} to the UE, whom has maximal upper bound among available UEs. Finally, it removes RB_{pick} from $\mathcal{M}$ to $\mathcal{M}'$, and if this allocation cause the UE, whom obtains $\mathrm{RB}_{scheduled}$, blocked, then removes the UE from $\mathcal{N}$ to $\mathcal{N}'$.

3.3.3. Detailed Example

A detailed example is illustrated in this paragraph, in which we assume the UL bandwidth is divided into 4 RBs (denoted as $\mathrm{RB}_1, \mathrm{RB}_2, \mathrm{RB}_3$ and RB_4) and 3 active UEs (denoted as $\mathrm{UE}_1, \mathrm{UE}_2$ and UE_3) want to transmit UL resources. After active UEs reporting their channel qualities to eNodeB, the eNodeB gets SNR values ($\delta_{i,j}$) of active UEs as shown in Table III.

Table III. Example – Initial SNR table

	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701

Let *window size* (*ws*) of *window* (*w*) be 3. At "Build window SNR table", *w* is placed on all the RBs of UEs of Table III to build *window table*. Here, we should note that since RB_1 is the first RB index, there is no RB on the left of RB_1. Thus, the *w* at RB_1 only covers RB_1 and RB_2. The value of $\mathrm{wSNR}_{1,1}$ is expressed as:

$$\mathrm{wSNR}_{1,1} = \min(\delta_{1,1}, \delta_{1,2}) = \min(3.945, 7.753) = 3.945$$

Similarly, since RB_4 is the last index, there is no RB on the right of RB_4. Thus, the *w* at RB_4 only covers RB_3 and RB_4. The value of $\mathrm{wSNR}_{1,4}$ is expressed as:

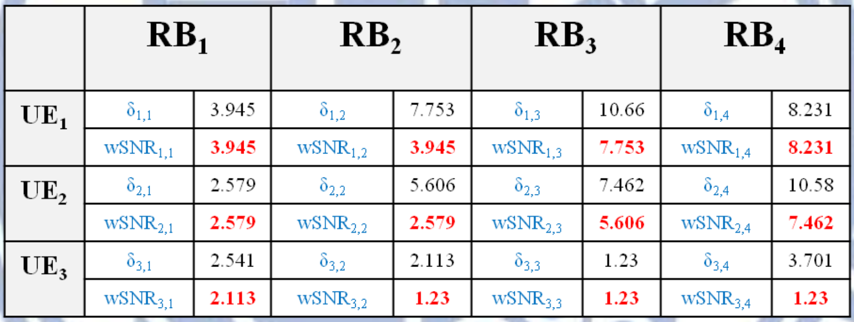
$$\mathrm{wSNR}_{1,4} = \min(\delta_{1,3}, \delta_{1,4}) = \min(10.66, 8.231) = 8.231$$

For RB_2 and RB_3, there is RB both on left and right. The *w* can hence properly cavers RB_1, RB_2 and RB_3 for RB_2. For RB_3, the *w* can properly covers RB_2, RB_3 and RB_4. The value of $\mathrm{wSNR}_{1,2}$ and $\mathrm{wSNR}_{1,3}$ are expressed as:

$$\mathrm{wSNR}_{1,2} = \min\left(\delta_{1,1}, \delta_{1,2}, \delta_{1,3}\right) = \min(3.945, 7.753, 10.66) = 3.945$$
$$\mathrm{wSNR}_{1,3} = \min\left(\delta_{1,2}, \delta_{1,3}, \delta_{1,4}\right) = \min(7.753, 10.66, 8.231) = 7.753$$

The procedure of deriving $\mathrm{wSNR}_{\mathrm{i,j}}$ values of UE_2 and UE_3 from SNR values $(\delta_{i,j})$ is similar to UE_1. The result is shown in Table IV

Table IV. Example – Build window SNR table



	RB_1		**RB_2**		**RB_3**		**RB_4**	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
	$\mathrm{wSNR}_{1,1}$	**3.945**	$\mathrm{wSNR}_{1,2}$	**3.945**	$\mathrm{wSNR}_{1,3}$	**7.753**	$\mathrm{wSNR}_{1,4}$	**8.231**
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$\mathrm{wSNR}_{2,1}$	**2.579**	$\mathrm{wSNR}_{2,2}$	**2.579**	$\mathrm{wSNR}_{2,3}$	**5.606**	$\mathrm{wSNR}_{2,4}$	**7.462**
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$\mathrm{wSNR}_{3,1}$	**2.113**	$\mathrm{wSNR}_{3,2}$	**1.23**	$\mathrm{wSNR}_{3,3}$	**1.23**	$\mathrm{wSNR}_{3,4}$	**1.23**

At "Choose starting RB to UE", we first calculate Δ_1, Δ_2, Δ_3 and Δ_4, respectively. For RB_1, $\max^1\left(wSNR_{1,1}\right) = 3.945$ and $\max^2\left(wSNR_{2,1}\right) = 2.579$. Thus, Δ_1 can be calculated as:

$$\Delta_1 = \max^1\left(wSNR_{1,1}\right) - \max^2\left(wSNR_{2,1}\right) = 3.945 - 2.579 = 1.366$$

For RB_2, $\max^1\left(wSNR_{1,2}\right) = 3.945$ and $\max^2\left(wSNR_{2,2}\right) = 2.579$. Thus, Δ_2 can be calculated as:

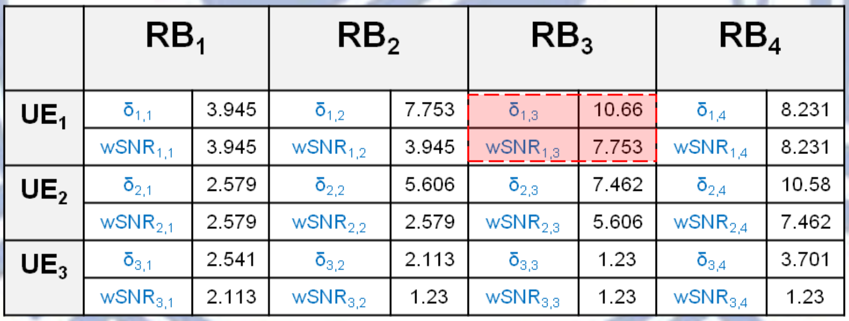
$$\Delta_2 = \max^1(wSNR_{1,2}) - \max^2(wSNR_{2,2}) = 3.945 - 2.579 = 1.366$$

For RB_3, $\max^1(wSNR_{1,3}) = 7.753$ and $\max^2(wSNR_{2,3}) = 5.606$. Thus, Δ_3 can be calculated as:

$$\Delta_3 = \max^1(wSNR_{1,3}) - \max^2(wSNR_{2,3}) = 7.753 - 5.606 = 2.147$$

The calculation of Δ_4 is similar to Δ_1, Δ_2 and Δ_3, which is 0.769. Since Δ_3 is the maximal, RB_3 is chosen as starting RB at this procedure. Besides, $\max^1(wSNR_{1,3})$ belongs to UE_1, we hence allocates RB_3 to UE_1. The result is shown in Table V, where the red grid on RB_3 of UE_1 manes the RB_3 allocated to UE_1.

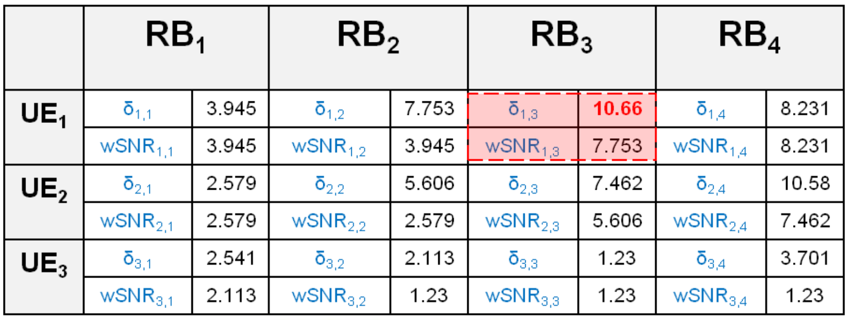
Table V. Example – Choose starting RB to UE



	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
	$\mathrm{wSNR}_{1,1}$	3.945	$\mathrm{wSNR}_{1,2}$	3.945	$\mathrm{wSNR}_{1,3}$	7.753	$\mathrm{wSNR}_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$\mathrm{wSNR}_{2,1}$	2.579	$\mathrm{wSNR}_{2,2}$	2.579	$\mathrm{wSNR}_{2,3}$	5.606	$\mathrm{wSNR}_{2,4}$	7.462
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$\mathrm{wSNR}_{3,1}$	2.113	$\mathrm{wSNR}_{3,2}$	1.23	$\mathrm{wSNR}_{3,3}$	1.23	$\mathrm{wSNR}_{3,4}$	1.23

Since the latest scheduling allocates RB_3 to UE_1, $\delta_{1,3}$ is get at “Adjust SNR table”. In this procedure, SNR values of UE_1 are checked. If the checked SNR value is larger than $\delta_{1,3}$, the SNR value is assigned to value of $\delta_{1,3}$. From observation, we know value of $\delta_{1,1}$, $\delta_{1,2}$ and $\delta_{1,4}$ is less than $\delta_{1,3}$. Thus, the SNR value doesn’t need to revise. The result is shown in Table VI, where the red bold word 10.66 is $\delta_{1,3}$.

Table VI. Example – Adjust SNR table



	RB_1		**RB_2**		**RB_3**		**RB_4**	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	**10.66**	$\delta_{1,4}$	8.231
	$wSNR_{1,1}$	3.945	$wSNR_{1,2}$	3.945	$wSNR_{1,3}$	7.753	$wSNR_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$wSNR_{2,1}$	2.579	$wSNR_{2,2}$	2.579	$wSNR_{2,3}$	5.606	$wSNR_{2,4}$	7.462
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$wSNR_{3,1}$	2.113	$wSNR_{3,2}$	1.23	$wSNR_{3,3}$	1.23	$wSNR_{3,4}$	1.23

At “Adjust window SNR table”, RB_2 is first on the left side of allocated RBs. Since the right side of RB_2 is allocated to UE_1 and $3 \geq 3 = 2 - \frac{3-1}{2} + 3 - 1$, UE_1 doesn’t need to revise left side unchecked RBs. For UE_2, new $\mathrm{wSNR}_{2,2}$ is calculated as:

$$\mathrm{wSNR}_{2,2} = \min(\delta_{2,1}, \delta_{2,2}) = \min(2.579, 5.606) = 2.579$$

Since the new $\mathrm{wSNR}_{2,2}$ equals to original one, $\mathrm{wSNR}_{2,2}$ doesn’t need to revise. Besides, the stop determination happens at RB_2, we don’t need to check RB_1 and $\mathrm{wSNR}_{2,1}$. For UE_3, new $\mathrm{wSNR}_{3,2}$ is calculated as:

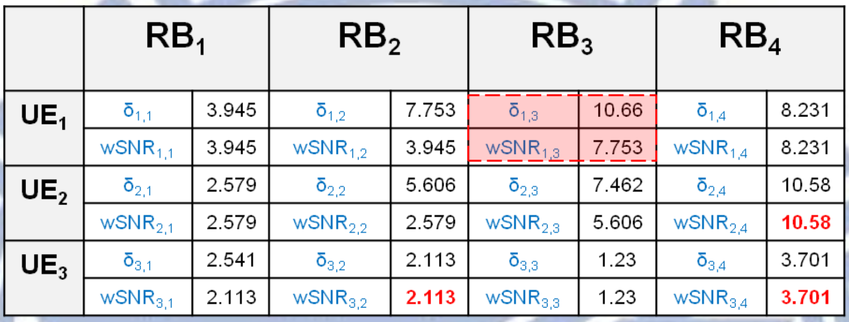
$$\mathrm{wSNR}_{3,2} = \min(\delta_{3,1}, \delta_{3,2}) = \min(2.541, 2.113) = 2.113$$

Since the new $\mathrm{wSNR}_{3,2}$ not equals to original one, which is 1.23, $\mathrm{wSNR}_{3,2}$ is adjusted to 2.113. Similarly, the stop determination happens at RB_2, we don’t need to check RB_1 and $\mathrm{wSNR}_{3,1}$. For right side unchecked RBs, RB_4 is first on the right side of allocated RBs. Since the left side of RB_4 is allocated to UE_1 and $3 \leq 3 = 4 - \frac{3-1}{2}$, UE_1 doesn’t need to revise right side unchecked RBs. For UE_2, new $\mathrm{wSNR}_{2,4}$ is calculated as:

$$\mathrm{wSNR}_{2,4} = \min(\delta_{2,4}) = \min(10.58) = 10.58$$

Since the new $\mathrm{wSNR}_{2,4}$ not equals to original one, which is 7.462, $\mathrm{wSNR}_{2,4}$ is adjusted to 10.58. The procedure of UE_3 is similar to UE_2, and $\mathrm{wSNR}_{3,4}$ is adjusted to 3.701. The result of this procedure shows in Table VII, in which the red bold words are revised values.

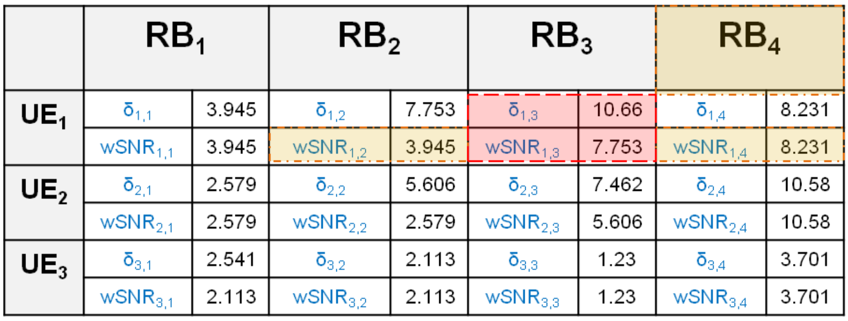
Table VII. Example – Adjust window SNR table



	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
	$wSNR_{1,1}$	3.945	$wSNR_{1,2}$	3.945	$wSNR_{1,3}$	7.753	$wSNR_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$wSNR_{2,1}$	2.579	$wSNR_{2,2}$	2.579	$wSNR_{2,3}$	5.606	$wSNR_{2,4}$	**10.58**
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$wSNR_{3,1}$	2.113	$wSNR_{3,2}$	**2.113**	$wSNR_{3,3}$	1.23	$wSNR_{3,4}$	**3.701**

At “Choose next RB” procedure, RB_2 is RB_{left} and RB_4 is RB_{right}. Due to the right side of RB_{left} is RB_3, which is allocated to UE_1, the decision value of RB_{left} is $\mathrm{wSNR}_{1,2} = 3.945$. On the other hand, the left side of RB_{right} is also RB_3, the decision value of RB_{right} is $\mathrm{wSNR}_{1,4} = 8.231$. Since $8.231 > 3.945$, RB_{right} (or RB_4) is chosen as RB_{pick}. The result is shown in Table VIII, where the yellow grid on RB_4 means RB_4 is chosen to be checked.

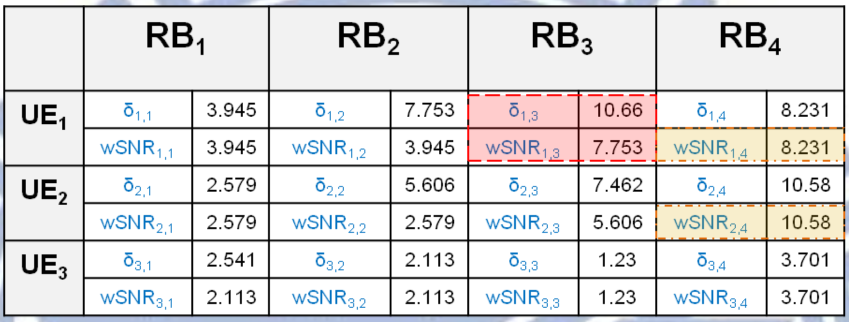
Table VIII. Example – Choose next RB



	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
	$wSNR_{1,1}$	3.945	$wSNR_{1,2}$	3.945	$wSNR_{1,3}$	7.753	$wSNR_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$wSNR_{2,1}$	2.579	$wSNR_{2,2}$	2.579	$wSNR_{2,3}$	5.606	$wSNR_{2,4}$	10.58
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$wSNR_{3,1}$	2.113	$wSNR_{3,2}$	2.113	$wSNR_{3,3}$	1.23	$wSNR_{3,4}$	3.701

At “Choose available allocated UEs”, $\mathrm{RB}_{scheduled}$(or RB_3) is not allocated to virtual UE. Thus the condition (1) of this procedure is skipped. Since $\mathrm{wSNR}_{2,4}$ (or $\mathrm{wSNR}_{2,pick}$) is larger than $\mathrm{wSNR}_{1,4}$ (or $\mathrm{wSNR}_{1,pick}$), condition (2) is met, where $\mathrm{UEs}_{available} = \{\mathrm{UE}_1, \mathrm{UE}_2, \mathrm{vUE}\}$. The result shows in Table IX, where the yellow grids on RB_4 of UE_1 and RB_4 of UE_2 means UE_1 and UE_2 are selected in $\mathrm{UEs}_{available}$ set.

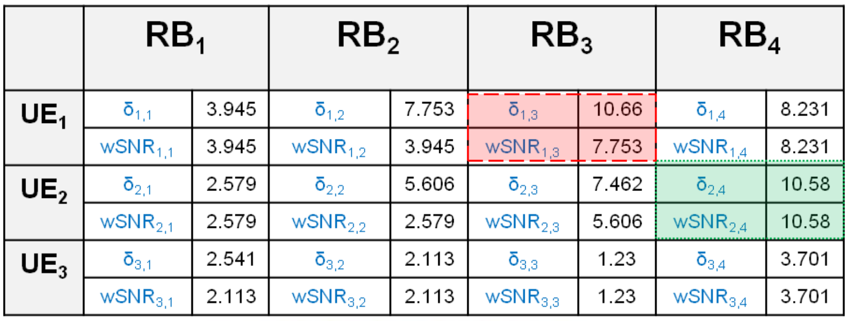
Table IX. Example – Choose available allocated UEs



	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
	$\mathrm{wSNR}_{1,1}$	3.945	$\mathrm{wSNR}_{1,2}$	3.945	$\mathrm{wSNR}_{1,3}$	7.753	$\mathrm{wSNR}_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$\mathrm{wSNR}_{2,1}$	2.579	$\mathrm{wSNR}_{2,2}$	2.579	$\mathrm{wSNR}_{2,3}$	5.606	$\mathrm{wSNR}_{2,4}$	10.58
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$\mathrm{wSNR}_{3,1}$	2.113	$\mathrm{wSNR}_{3,2}$	2.113	$\mathrm{wSNR}_{3,3}$	1.23	$\mathrm{wSNR}_{3,4}$	3.701

At “Upper bound estimation”, for UE_1, $\mathrm{UEs}_{left} = \{\mathrm{UE}_1, \mathrm{UE}_2, \mathrm{UE}_3, \mathrm{vUE}\}$ and $\mathrm{UEs}_{right} = \emptyset$. Since the decision value of RB_2, which is $\delta_{1,2} = 7.753$, is larger than RB_1, which is $\delta_{1,1} = 3.945$, the sorting list of UE_1 is $\{\mathrm{RB}_2, \mathrm{RB}_1\}$. At first, RB_4 is pseudo-allocated to UE_1, and gets 16.462. Then the upper bounds after pseudo-allocating RB_2 to UE_1, UE_2 UE_3 or vUE are 23.259, 22.068, 18.575 and 16.462, respectively. Since the upper bound after pseudo-allocating RB_2 to UE_1 is maximal, RB_2 is pseudo-allocated to UE_1 and gets 23.259. Similarly, for RB_1, the upper bounds after pseudo-allocating RB_1 to UE_1, UE_2 UE_3 or vUE are 15.78, 25.838, 25.8 and 23.259, respectively. Since 25.838 is maximal and no RBs need to calculate, the upper bound estimation of allocating RB_4 to UE_1 is 25.838. The calculations of allocating RB_4 to UE_2 or vUE are similar to above, in which the estimation of UE_2 is 28.627. Besides, the estimation of vUE is 18.807. Due to the upper bound estimation of UE_2 is larger than UE_1 and vUE, which is 28.627, thus we allocates RB_4 to UE_2. The result shows in Table X, where the green grid on RB_4 of UE_2 manes UE_2 obtains RB_4.

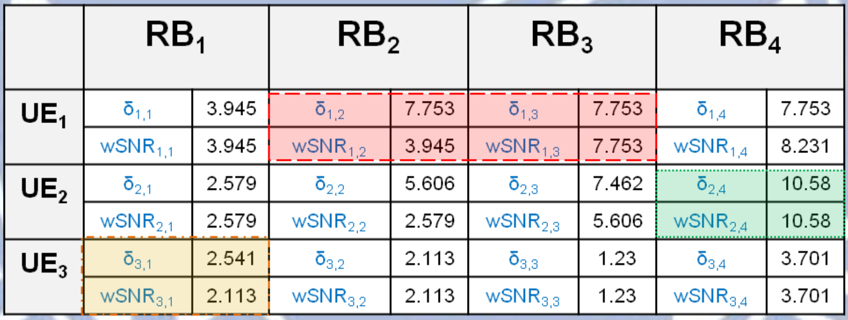
Table X. Example – Upper bound estimation



	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	10.66	$\delta_{1,4}$	8.231
	$wSNR_{1,1}$	3.945	$wSNR_{1,2}$	3.945	$wSNR_{1,3}$	7.753	$wSNR_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$wSNR_{2,1}$	2.579	$wSNR_{2,2}$	2.579	$wSNR_{2,3}$	5.606	$wSNR_{2,4}$	10.58
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$wSNR_{3,1}$	2.113	$wSNR_{3,2}$	2.113	$wSNR_{3,3}$	1.23	$wSNR_{3,4}$	3.701

The rest of this example follows the flowchart in Fig. 9. The scheduling result is shown in Table XI, where $\{RB_1\}$ is allocated to UE_3, $\{RB_2, RB_3\}$ are allocated to UE_1, and $\{RB_4\}$ is allocated to UE_2.

Table XI. Example – Scheduling result



	RB_1		RB_2		RB_3		RB_4	
UE_1	$\delta_{1,1}$	3.945	$\delta_{1,2}$	7.753	$\delta_{1,3}$	7.753	$\delta_{1,4}$	7.753
	$wSNR_{1,1}$	3.945	$wSNR_{1,2}$	3.945	$wSNR_{1,3}$	7.753	$wSNR_{1,4}$	8.231
UE_2	$\delta_{2,1}$	2.579	$\delta_{2,2}$	5.606	$\delta_{2,3}$	7.462	$\delta_{2,4}$	10.58
	$wSNR_{2,1}$	2.579	$wSNR_{2,2}$	2.579	$wSNR_{2,3}$	5.606	$wSNR_{2,4}$	10.58
UE_3	$\delta_{3,1}$	2.541	$\delta_{3,2}$	2.113	$\delta_{3,3}$	1.23	$\delta_{3,4}$	3.701
	$wSNR_{3,1}$	2.113	$wSNR_{3,2}$	2.113	$wSNR_{3,3}$	1.23	$wSNR_{3,4}$	3.701

3.3.4. Pseudo Code and Time Complexity

To help analysis time complexity of proposed algorithm – UBERA, we present the pseudo codes here. In Table XII., we list the main body. Besides, Table XIII and Table XIV list the "Adjust SNR table" and "Adjust window SNR table" procedure, respectively.

In Table XII, line 1 to line 16 execute "Build window SNR table", which build a brand-new table based on SNR values, where eNodeB get from UEs' periodically report. Here, the SNRs in window size of UE at specific RB are all checked at line 8, and find the minimum among these SNRs as corresponding wSNR at line 9. From line 18 to line 35, “Choose starting RB to UE” is executed, where we get all the Δj from line 24 to line 29. At line 30, we check whether there are Δjs having same value as $max\Delta j$. The RB decision and allocation is from line 31 to line 35. At line 37, “Adjust SNR table” is executed and listed in Table XIII. In Table XIII, SNR of RB of specific UE is got at line 5, checked at line 6 and assigned at line 7. At line 38 of Table XII, “Adjust window SNR table” is executed and listed in Table XIV. In Table XIV, the searching order of RBs, which are in window size of left side non-checked RBs, is decided at line 7 and line 8. The searching order of RBs, which are in window size of right side non-checked RBs, is decided at line 9 and line 10. Besides, the new wSNR is continued updated from line 12 to line 19, and revise the old wSNR at line 21.

At line 40, algorithm goes into while-loop for allocating remaining non-checked RBs. From line 42 to line 70, “Choose next RB” is executed to pick one of non-checked RBs. If no RB needs to check, we return “no remaining RB needs to be checked” at line 47. The decision value of RB_{left} and RB_{right} got from line 55 to line 59 and from line 60 to line 64, respectively. From line 65 to line 69, RB to be checked is chosen. If “no remaining RB needs to be checked” is returned, the algorithm is terminated at line 73. Otherwise, “Choose available allocated UEs” is executed to choose candidate UEs from line 76 to line 83. From line 85 to line 100, “Upper bound estimation” is executed, in which the upper bound value of UE is calculated at line 93 according to 3.3.2 (7), and the decision of $maxUB_{i'}$ is from line 94 to line 97. In line 102, “Adjust SNR table” is executed to Table XIII. Besides. “Adjust window SNR table” is also executed to Table XIV.

Table XII. Pseudo code of UBERA

```
1:  // Build window SNR table
2:  Let ws be the size of the window
3:  Let $\delta_{min}$ be the minimum SNR in ws
4:  for each UE i in the system do
```

```
    for each RB j in the system do
        $\delta_{min}$ ← infinitely large;
        for each RB c in the ws of j do
            Let $\delta_{i,c}$ be the SNR of UE i on RB c
            if (c is in the legal range of RB) and ($\delta_{i,c}$ is less than $\delta_{min}$) then
                $\delta_{min} \leftarrow \delta_{i,c}$ ;
            end if
        end for
        Let $wSNR_{i,j}$ be the window SNR of UE i on RB j
        $wSNR_{i,j} \leftarrow \delta_{min}$ ;
    end for
end for

// Choose starting RB to UE
Let $wSNR_{i,j}^{(1)}$ be the 1st high window SNR on RB j, which belongs to UE i
Let $wSNR_{i',j}^{(2)}$ be the 2nd high window SNR on RB j, which belongs to UE i'
Let Δj be the difference between $wSNR_{i,j}^{(1)}$ and $wSNR_{i',j}^{(2)}$ at RB j
Let maxΔj be the maximum of all the Δj
maxΔj ← -1;
for each RB j in the system do
    Δj ← $wSNR_{i,j}^{(1)} - wSNR_{i',j}^{(2)}$ ;
    if Δj is larger than maxΔj then
        maxΔj ← Δj;
    end if
end for
Check if other RB j' has the same Δj' as maxΔj, where j' is not equal to j
if doesn't have then
    Allocate RB j of maxΔj to UE i
else
    Allocate RB j to UE i, who has the same Δj as maxΔj and $wSNR_{i,j}^{(1)}$ is the maximum
end if

Execute Adjust SNR table [Table V.];
Execute Adjust window SNR table [Table VI.];

while (true) do

    // Choose next RB
```

```
Let $RB_{left}$ be the first un-checked RB index on the left side of already checked RBs
Let $RB_{right}$ be the first un-checked RB index on the right side of already checked RBs
Let $RB_{pick}$ be the chosen RB here
if both $RB_{left}$ and $RB_{right}$ doesn't exists then
    $RB_{pick} =$ no remaining RB needs to be checked
else if $RB_{left}$ doesn't exist then
    $RB_{pick} = RB_{right}$
else if $RB_{right}$ doesn't exist then
    $RB_{pick} = RB_{left}$
else
    Let $wSNR_{left}$ be the decision value of $RB_{left}$
    Let $wSNR_{right}$ be the decision value of $RB_{right}$
    if right side of $RB_{left}$ is allocated to virtual UE then
        $wSNR_{left} \leftarrow$ maximum window SNR of UEs, who is allowed to be allocated at
        $RB_{left}$
    else
        $wSNR_{left} \leftarrow$window SNR of UE at $RB_{left}$, who is already allocated at right side of
        $RB_{left}$
    end if
    if left side of $RB_{right}$is allocated to virtual UE then
        $wSNR_{right} \leftarrow$ maximum window SNR of UEs, who is allowed to be allocated at
        $RB_{right}$
    else
        $wSNR_{right} \leftarrow$window SNR of UE at $RB_{right}$, who is already allocated at left side of
        $RB_{right}$
    end if
    if $wSNR_{left}$ is larger than $wSNR_{right}$ then
        $RB_{pick} = RB_{left}$
    else
        $RB_{pick} = RB_{right}$
    end if
end if

if $RB_{pick} ==$ no remaining RB needs to be checked then
    break;
end if

// Choose available allocated UEs
```

```
    Let $c$ be $RB_{pick}$
    Let $wSNR_c$ be the window SNR of UE $u$ at $c$, which is already allocated at the side of $c$
    if ($u$ is real UE) and (there exists other UE allowed to be allocated at $c$, and has higher
      window SNR than $wSNR_c$) then
        Let $I$ be the set of UEs allowed to be allocated at $c$, whom has higher window SNR than
         $wSNR_c$
    else if ($u$ is virtual UE) or (($u$ is real UE) and (allocate $c$ to $u$ will cause the total system
          throughput down)) then
        Let $I$ be the set of UEs allowed to be allocated at $c$
    end if

    // Upper bound estimation
    if $I$ is empty then
        allocate $c$ to $u$
    else
        Let $UB_i$ be the upper bound value of UE $i$
        Let $maxUB_{i'}$ be the maximum among all the $UB_i$s, which belong to UE $i'$
        $maxUB_{i'}$ = -1;
        for each UE $i$ in $I$ do
            Calculate $UB_i$ according to 3.3.2(7)
            if $UB_i$ is larger than $maxUB_{i'}$ then
                $maxUB_{i'} \leftarrow UB_i$
                $i' \leftarrow i$;
            end if
        end for
        allocate $c$ to $i'$
    end if

    Execute Adjust SNR table [Table V.];
    Execute Adjust window SNR table [Table VI.];
end while
```

Table XIII. Pseudo code of Adjust SNR table

```
Let $c$ be the latest scheduled RB, which is allocated to UE $i$
if UE $i$ is not a virtual UE then
    Let $\delta_{i,c}$ be the SNR of UE $i$ on RB c
```

```
    for each RB $j$ in the system except $c$ do
        Let $\delta_{i,j}$ be the SNR of UE $i$ on RB $j$
        if $\delta_{i,j}$ is larger than $\delta_{i,c}$ then
            $\delta_{i,j} \leftarrow \delta_{i,c}$ ;
        end if
    end for
end if
```

Table XIV. Pseudo code of Adjust window SNR table

```
 1: Let i' be the latest allocated UE
 2: Let ws be the size of the window
 3: Let $\delta_{min}$ be the minimum SNR in ws
 4: for each UE i in the system do
 5:     for each RB j in the system do
 6:         $\delta_{min}$ = infinitely large;
 7:         if (j is not allocated) and (is at left side of allocated RBs) then
 8:             check RB c form low to high index at 12:
 9:         else if (j is not allocated) and (is at right side of allocated RBs) then
10:             check RB c from high to low index at 12:
11:         end if
12:         for each RB c in the ws of j do
13:             Let $\delta_{i,c}$ be the SNR of UE i on RB c
14:             if (((i is not i') and (c is not allocated to any UE)) or ((i is i') and (c is not allocated to any UE
                    or allocated to i'))) and (c is in the legal range of RB) and ($\delta_{i,c}$ is less than $\delta_{min}$) then
15:                 $\delta_{min} \leftarrow \delta_{i,c}$;
16:             else if ((i is not i') and (c is allocated to UE)) or ((i is i') and (c is allocated to UE but not i'))
17:                 break;
18:             end if
19:         end for
20:         Let $wSNR_{i,j}$ be the window SNR of UE i on RB j
21:         $wSNR_{i,j} \leftarrow \delta_{min}$
22:     end for
23: end for
```

At “Build window SNR table”, it checks all RBs of all UEs to build the *window table*. Since the size of w is ws, it takes $O(n \cdot m \cdot ws)$. At “Choose starting RB to UE”, the scheduler first checks all RBs of UEs to get $\max^1\left(wSNR_{i,j}\right)$ and $\max^2\left(wSNR_{i,j}\right)$ of all RBs, and calculates Δ_j. Thus it takes $O(n \cdot m)$. Finding maximal Δ_j, checking same maximal Δ_j, and allocating to UE only needs to check all the Δ_j. It here takes $O(m)$. Thus, the complexity at “Choose starting RB to UE” takes $O(n \cdot m + m) = O(n \cdot m)$. Besides, “Adjusts SNR table” revises all $\delta_{i,j}$ of a specific UE$_i$, thus it takes $O(m)$ scanning all the RBs. Let the number of RBs be checked currently as c. “Adjust window SNR table” revises *window table* similar to “Build window SNR table”, but doesn’t revises the checked RBs. Hence, the scheduler takes $O(n \cdot (m - c) \cdot ws)$. Since RB$_{left}$ and RB$_{right}$ are updated after each allocation, it doesn’t take any time to find the two RBs. The worst case of “Choose next RB” is that RB$_{left+1}$ and RB$_{right-1}$ both are allocated to virtual UE. Thus, it takes $O(n)$ to find the decision values of RB$_{left}$ and RB$_{right}$ from $\mathcal{N}$. In the worst case of “Choose available allocated UEs”, all the UEs have the maximal $wSNR_{i,j}$ at RB$_{pick}$ among all the RBs of the corresponding UE. Thus, it takes $O(n \cdot m)$ to find the second maximal $wSNR_{i,j}$ and calculates Δ_{gain}. At “Upper bound estimation”, the worst case is that all the UEs belong to UEs$_{available}$, UEs$_{left}$ and UEs$_{right}$. It hence takes $O\left(n \cdot \left((m - c - 1) \cdot n \cdot m\right)\right) = O(n^2 \cdot (m - c - 1) \cdot m)$. The outside n means the number of UEs$_{available}$. $(m - c - 1)$ is the number of unallocated RBs, which needs to predict but doesn’t include RB$_{pick}$, since RB$_{pick}$ is pseudo-allocated at outside n. The inside n is the number of UEs$_{left}$ or UEs$_{right}$. Finally, the inside m is used to pseudo-adjust $\delta_{i,j}$, which is similar to “Adjust SNR table” and calculates the corresponding upper bound after pseudo-allocating to UEs$_{left}$ or UEs$_{right}$. So the total time complexity of the proposed method can be calculated as

$$
\begin{aligned}
&O(n\cdot m\cdot ws)+O(n\cdot m)+\\
&\sum_{c=1}^{m}\left(O(m)+O(n\cdot(m-c)\cdot ws)\right)+\\
&\sum_{c=1}^{m-1}\left(O(n)+O(n\cdot m)+O(\,n^2\cdot(m-c-1)\cdot m)\right)\\
&=O(n\cdot m\cdot ws)+O(n\cdot m)+O(n\cdot m^2\cdot ws)+O(n^2\cdot m^3)\\
&=O(n^2\cdot m^3)
\end{aligned}
$$

Since “Adjust SNR table” and “Adjust window SNR table” are called after each resource allocation, these two procedures run m times, respectively. Besides, since the first RB is allocated at “Choose starting RB to UE”, the “Choose next RB”, “Choose available allocated UEs” and “Upper bound estimation” run (m-1) times.

Chapter 4 Simulation Results

In this chapter, we first evaluate and compare the system throughput of proposed heuristic algorithms with optimal solution (denoted as OPT), Regular-CDS (denotes as CAS) [17] and two Regular-CDS based Smart-CDSs (denotes as TTRA and STRA [22]). Due to the exponential time complexity, OPT is only included into comparison with UEs vary from 1 to 10 where the UL bandwidth 3MHz is divided into 15 RBs. Then, we simulate large scale condition with UEs vary from 1 to 60 where the UL bandwidth 20 MHz is divides into 100 RBs. In this large scale simulation, OPT is not compared due to the problem of time complexity. Besides, two types of fading channel are also employed to evaluate the system throughput in different weather conditions and with physical objects between UE and eNodeB, in which the UEs vary from 1 to 30 with UL bandwidth 10MHz is divided into 50RBs. At second paragraph, we assume there are many active UEs want to transmit UL resources, in which the RBs are less than active UEs. In this simulation, the starvation ratio is proposed to evaluate how many UEs don't grant RBs for UL transmission. We compare the starvation ratio of proposed method with CAS, TTRA and STRA. To observe the influence of *window size*, we compare the system throughput of proposed algorithm itself by adjusting *window size* at third paragraph. The problem of proposed method is discussed at final paragraph. In system bandwidth, the UL bandwidth is set to 3MHz, 10MHz and 20MHz with 15 RBs, 50 RBs and 100 RBs, respectively. Each RB in frequency is composed of 12 subcarriers with 15 kHz per subcarrier. In time domain, each RB consists 1ms which is composed of 7 OFDM symbols. The simulation result is the average of 1,000 ms, where we assume the active UEs report there channel conditions every 1ms. According to specification, LTE offers 8 different MCSs for modulating UEs' transmission data. The rest of parameter settings are listed in Table XV.

Table XV. Parameter settings of the LTE UL system

Parameter	Setting
System bandwidth	3 MHz, 10 MHz, 20 MHz
Subcarriers per RB	12
Symbols per subcarrier	7
Bandwidth of RB	180 kHz
Number of RBs	15, 50, 100
Number of active UEs	1 ~ 10, 1 ~ 30, 1 ~ 60
Location of UE	random
Transmission Time Interval (TTI)	1 ms
Simulation time	1000 TTIs
Modulation and Coding Scheme	QPSK (1/2、2/3、3/4) 16 QAM (1/2、2/3、3/4) 64 QAM (2/3、3/4)
Fading channel (μ)	Frequency-selective/Flat fading
Window size (*ws*)	1, 5, 9, 13

4.1 System Throughput

In Fig. 10, the number of deployed UEs varies from 1 to 10 with 15 RBs per TTI. In this figure, we can observe that the optimum solution performs the best among the rest approaches. However, UBERA always performs better than CAS, TTRA and STRA, and is much closer to optimum. Since constraint of robust MCS mode is not considered in CAS, it performs worst among the compared methods. Although robust MCS mode is considered in TTRA and STRA, these methods are all proposed based on CAS. Thus, the grown up of these two is limited by CAS.

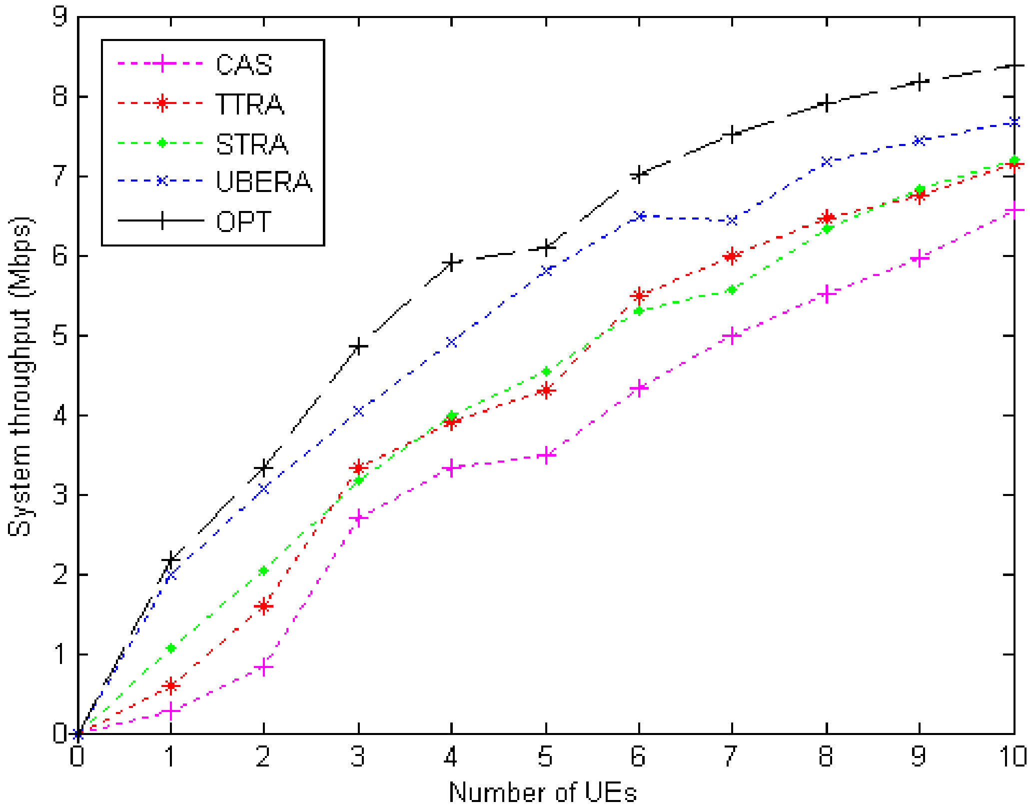


Figure 10. System throughput with optimum solution be compared

In Fig. 11, the number of deployed UEs varies from 1 to 60 with 100 RBs per TTI. Since the exponential time complexity of optimum solution, we only compare UBERA with CAS, TTRA and STRA. From the simulation result, we can observe that UBERA outperforms than the other three approaches. However, as the number of UEs higher than 45, the difference between UBERA and the other three approaches decreases. The reason is that, as the number of UEs grows to a degree, the maximal system throughput approaches. Thus the growing of system throughput becomes slowly. Besides, because the multi-user diversity gains of LTE UL, the throughput of all these approaches grows as number of UEs increases.

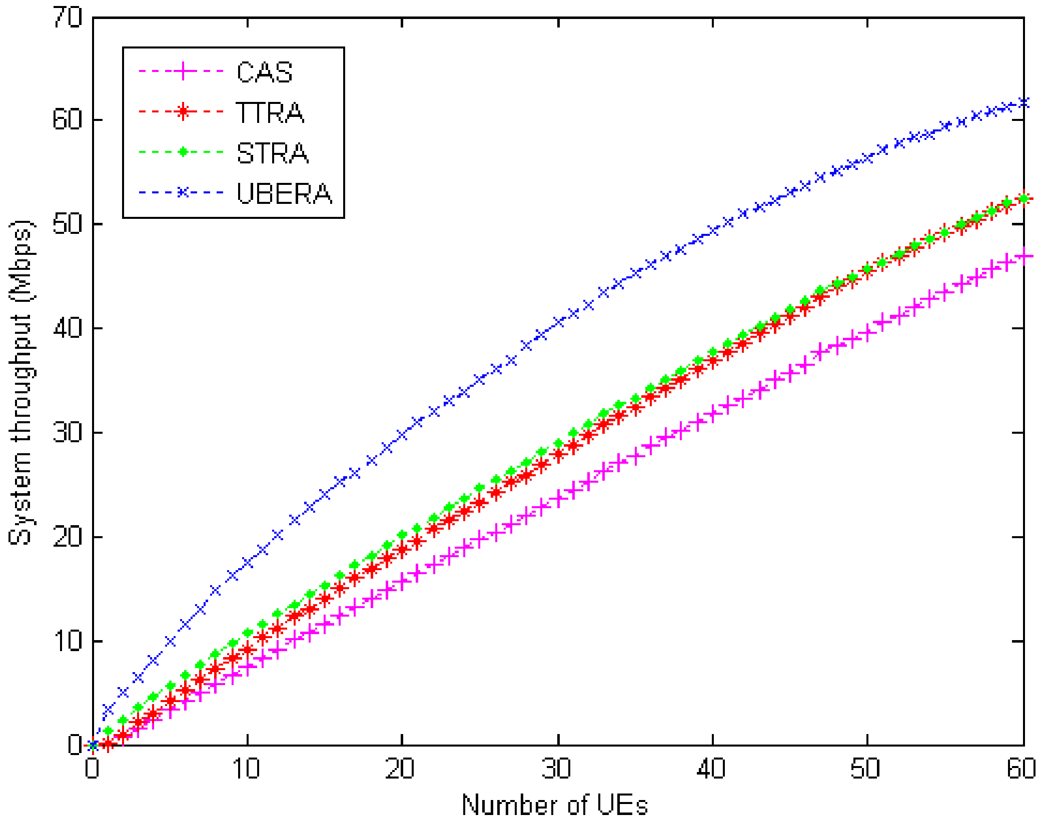


Figure 11. System throughput without optimum solution be compared

Then, we compare the system throughput performance in more comprehensive situation. The results are shown in Fig. 12, in which the number of deployed UEs varies from 1 to 30 with 50 RBs per TTI. There are two types of channel are employed: frequency-selective fading, which can be regarded as an independent fading channel, denoted as$\mu = 0$; flat fading, which can be seen as a highly correlated fading channel, denoted as$\mu = 1$. In Fig. 12, UBERA always performs better than CAS, TTRA and STRA in both frequency-selective fading ($\mu = 0$) and flat fading ($\mu = 1$). Through observation, we know the proposed algorithm can still keep well performance in system throughput no matter the weather conditions and buildings between UE and eNodeB.

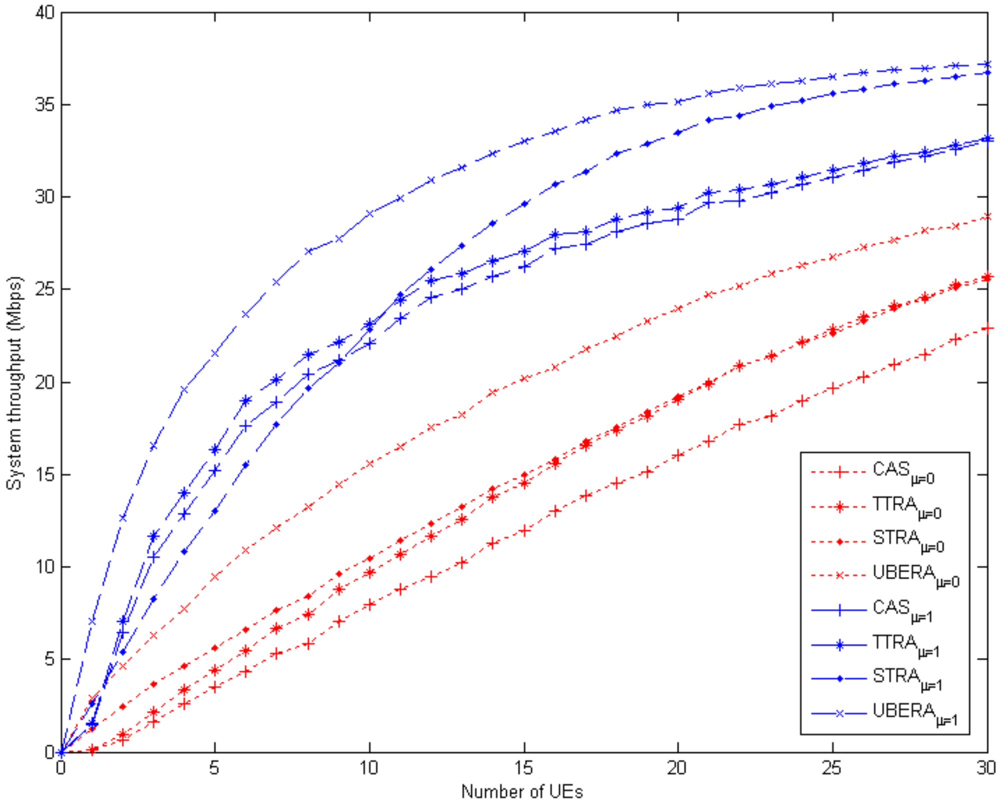


Figure 12. System throughput with different fading channel

4.2 Starvation Ratio

Due to the scarcity of wireless resources (or RBs) and there are many active UEs want to transmit UL resources, some active UEs may unable to grant RBs. Moreover, if the channel qualities of these active UEs are poor often, they will suffer from no more RBs to use. Hence, the starvation ratio (σ) is introduced to analysis the ratio of UEs unable to obtain RBs, where σ is expressed as:

$$\sigma = \frac{\sum_{i=1}^{n} \arg\left(|S_i(t)| = 0\right)}{n}$$

where numerator is number of UEs doesn’t obtain any RB at TTI t and denominator n is number of active UEs want to transmit UL resource.

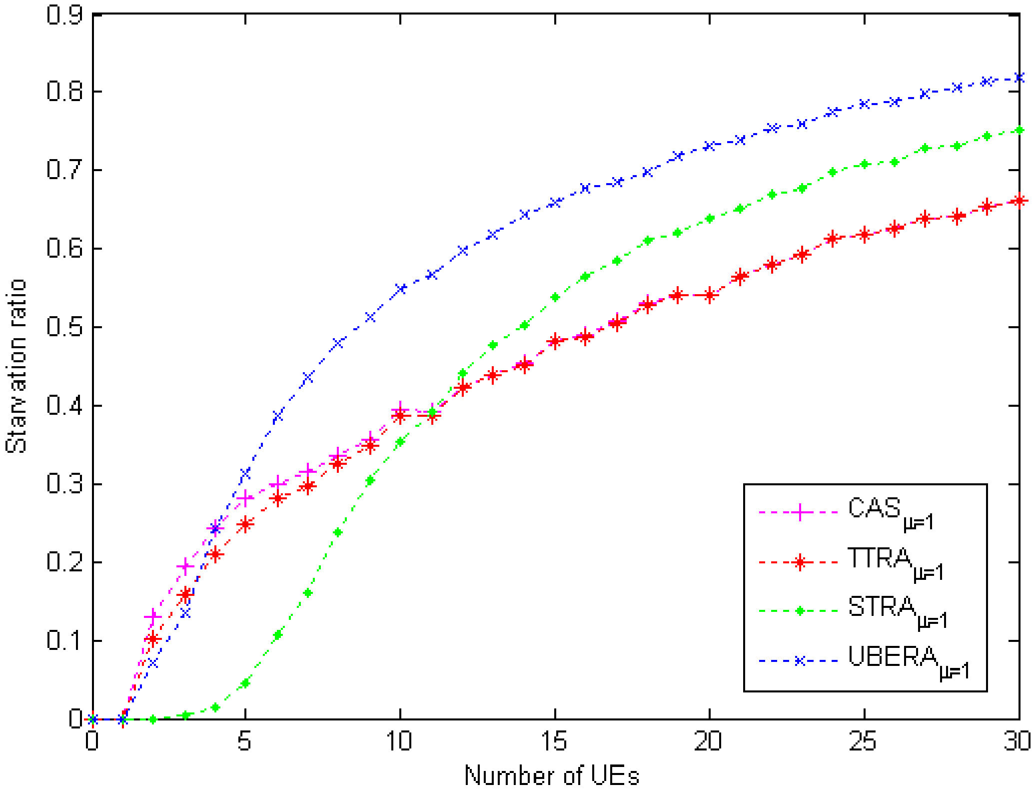


Figure 13. Starvation ratio vs. Number of UEs

The simulation result shows in Fig. 13, in which the number of deployed UEs varies from 1 to 30 with only 15 RBs per TTI. From simulation result, we observe the starvation ratio of UBERA is less than CAS and TTRA and higher than STRA while the number of UEs is less, and higher than all the three methods while the number of UEs is more than 5. Besides, by taking Fig. 12 and Fig. 13 into analysis, the system throughput of STRA in flat fading ($\mu = 1$) is lower than CAS and TTRA while UEs are less than 11. In addition, we can also observe the starvation ratio of STRA is lower than CAS and TTRA while UEs are less than 11. Thus, we can conclude there is tradeoff between system throughput and starvation ration. In other words, the better performance in system throughput means some active UEs owning well channel qualities obtain more RBs, which makes the UEs owning worse channel qualities unable to obtain any RB. Therefore, the performance of starvation ratio performs worse. Since the objective of proposed UBERA algorithm is to maximize system throughput, the simulation result of outage ratio doesn't perform well. However, the UBERA always performs better than CAS, TTRA and STRA in system throughput.

4.3 Influence of Window Size

In Fig. 14, we compare UBERA itself with different number of window size. The number of deployed UEs varies from 1 to 30 with 50 RBs per TTI. The observation shows that as the window size sets to 1, which means $\delta_{i,j} = wSNR_{i,j}$, it doesn't get well result. The best result is acquired as window size set to 5. We can also know that the higher window size cannot guarantee better result.

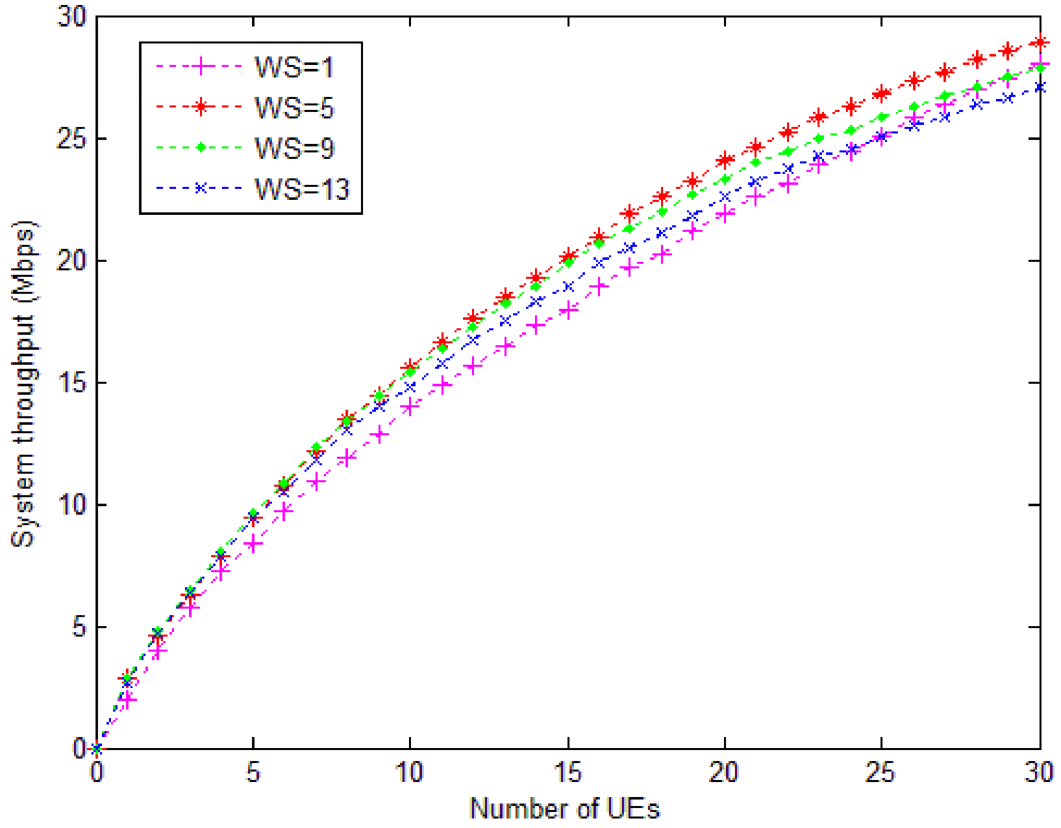
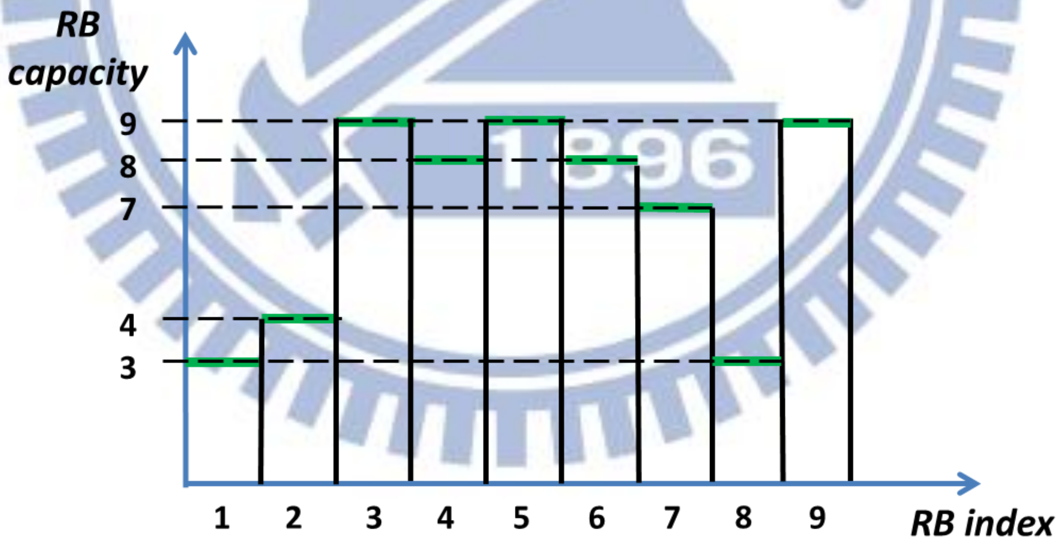


Figure 14. System throughput with different window size

An example shows in Fig. 15. As the window size set to 5, the scheduler can appropriately calculate $wSNR_{i,j}$ of the UE in this scenario. Hence, the scheduler can clearly notice that the UE's most advantageous contiguous RBs region is nearing RB_5. Thus we can try to allocate RBs nearing RB_5 to this UE. However, as the window size set too small such as 1, the scheduler will confuse whether it should allocate contiguous RBs nearing RB_3, RB_5 or RB_9 to the UE. If the scheduler decides to allocate RBs nearing RB_9, it will hence get low system throughput, because RB_8 doesn't have well capacity to the UE. In case we set the window size too large such as 9, $wSNR_{i,j}$ of the UE in this example are all equivalent. Thus, the scheduler can't make decision which RBs have abrupt well capacity to the UE. Since the RBs capacity conditions may different according to the positions of the UEs, applying different window size to UEs may be the best policy to the scheduler. Thus, dynamic adjusting window size will be one of our focused future works.



	RB_1	RB_2	RB_3	RB_4	RB_5	RB_6	RB_7	RB_8	RB_9
ws=1	3	4	9	8	9	8	7	3	9
ws=5	3	3	3	4	7	3	3	3	3
ws=9	3	3	3	3	3	3	3	3	3

Figure 15. An example of influence on different window size

4.4 Problem of UBERA

Some problems may cause the UBERA doesn’t work well. First, as the window size doesn’t suit in this scenario, the scheduler will make confuse as described above. Hence, it may pick RB to wrong UE compared to the allocation result of optimal solution at “Choose starting RB to UE”. For example, if window size set too large that all the UEs themselves have same $wSNR_{i,j}$ crossing all the RBs as in window size =9 of Fig. 15, we will get same Δ_j and $\max^1(wSNR_{i,j})$ at all the RBs. Thus it cannot accurately allocate suitable UE to RB at “Choose starting RB to UE”. Besides, since the step of “Upper bound estimation” owns the highest time complexity, the scheduler at “Choose available allocated UEs” tries to exclude some UEs from doing estimation, which may not be possible allocated at RB_{pick}. However, the determination cannot guarantee the scheduler will not exclude the UE of optimal solution at RB_{pick}. Thus it causes the system throughput goes down while excluding this UE. At last, if we try to get the upper bound of a specific UE allocated at RB_{pick} while considering constraints of LTE UL, it still is a NP-hard problem (LTE UL maximal system throughput while RB_{pick} is allocated to the specific UE). Thus, we don’t consider contiguous constraint (relax constraint) while doing estimation at step of “Upper bound estimation”, which will cause the estimation not so accurately every time and may allocate wrong UE to RB_{pick}.

Chapter 5 Conclusion

In this thesis, we first introduce two inherent constraints of SC-FDMA channel access scheme. Here we name these two constraints as contiguous RB assignment and robust MCS mode. Taking the two constraints into consideration, we formulate the scheduling problem of maximize LTE UL system throughput as Integer Linear Programming. Due to the exponential time complexity of optimal solution, we design an estimation based algorithm – Upper Bound Estimation Resource Allocation (UBERA). In simulation experiment, we compare UBERA with optimal solution, CAS, TTRA and STRA, The simulation results show the UBERA indeed have better performance than CAS, TTRA and STRA, while having fewer gaps from optimal in system throughput.

Since the window size is constant in single resource allocation period, our future work will focus on adjust window size automatically in one period to further enhance the performance of UBERA. Furthermore, “proportional fairness”, as known as long-term fairness, is also a critical issue in the LTE scheduling problem. By taking this issue into consideration, we will modify our proposed method to enhance the performance of proportional fairness while maintaining the system throughput at the same time.

References

[1] H. Ekstr¨om; A. Furusk¨ar; J. Karlsson; M. Meyer; S. Parkvall; J. Torsner; and M. Wahlqvist, “Technical Solutions for the 3G Long-Term Evolution,” IEEE Communications Magazine, vol. 44, no. 3, pp. 38–45, March 2006.

[2] Agilent Technologies, “Agilent 3GPP Long Term Evolution: System Overview, Product Development, and Test Challenges,” http://cp.literature.agilent.com/litweb/pdf/5989-8139EN.pdf, July 2009.

[3] Hyung G. Myung; Junsung Lim; David J. Goodman, “Single Carrier FDMA for Uplink Wireless Transmission,” IEEE Vehic. Tech. Mag., vol. 1, pp. 30-38, Sept. 2006.

[4] Freescale Semiconductor, "Long Term Evolution Protocol Overview," http://www.freescale.com/files/wireless_comm/doc/white_paper/LTEPTCLOVWWP.pdf, white paper, Document Number: LTEPTCLOVWWP, Rev 0, Oct. 2008.

[5] T. Lunttila; J. Lindholm; K. Pajukoski; E. Tiirola; and A. Toskala, “EUTRAN Uplink Performance,” International Symposium on Wireless Pervasive Computing (ISWPC) 2007, February 2007.

[6] Blogial NETWORK, “LTE – Long Term Evolution,” http://blogial.com/2009/09/20/lte-long-term-evolution/, Sep. 2009

[7] 3GPP TS 36.213 V10.5.0 “Physical layer procedures (Release 10),” http://www.3gpp.org/ftp/Specs/html-info/36213.htm, March 2012.

[8] 3GPP TR 25.814 V7.1.0, “Physical Layer Aspects for Evolved UTRA,” http://www.3gpp.org/ftp/Specs/html-info/25814.htm, Sept. 2006.

[9] Moray Rumney, “3GPP LTE: Introducing Single-Carrier FDMA,” http://cp.literature.agilent.com/litweb/pdf/5989-7898EN.pdf, Agilent Measurement Journal, 2008.

[10] Stefania Sesia; Issam Toufik; Matthew Baker, “LTE-the UMTS Long Term Evolution From Theory to Practice,” April 2009.

[11] Ruochen Wang; Zhiqiang He; Zheng Sun; Shan Lu; Kai Niu, “A Revenue-Based Low-Delay and Efficient Downlink Scheduling Algorithm in OFDMA Systems,” Vehicular Technology Conference (VTC) Fall, Sept. 2008.

[12] Fattah, H.; Alnuweiri, H., “A Cross-Layer Design for Dynamic Resource Block Allocation in 3G Long Term Evolution System,” Mobile Adhoc and Sensor Systems (MASS), Oct. 2009.

[13] Kwan, R.; Leung, C.; Jie Zhang, “Resource Allocation in an LTE Cellular Communication System,” International Conference on Communications (ICC), June 2009.

[14] Yong Li; Na Lu; Mugen Peng; Wenbo Wang, “Multiuser Resource Allocation for OFDM Downlink with Terminal Bandwidth Limitation,” Wireless Communications and Networking Conference (WCNC), April 2010.

[15] Yang, Xu ; Wang, Yapeng ; Zhang, Dapeng ; Cuthbert, Laurie, “Resource Allocation in LTE OFDMA Systems Using Genetic Algorithm and Semi-Smart Antennas,” Wireless Communications and Networking Conference (WCNC), April 2010.

[16] Calabrese, F.D.; Michaelsen, P.H.; Rosa, C.; Anas, M.; Castellanos, C.U.; Villa, D.L.; Pedersen, K.I.; Mogensen, P.E., “Search-Tree Based Uplink Channel Aware Packet Scheduling for UTRAN LTE,” Vehicular Technology Conference (VTC) Spring, May 2008.

[17] Ruiz de Temino, L.; Berardinelli, G.; Frattasi, S.; Mogensen, P, “Channel-aware scheduling algorithms for SC-FDMA in LTE uplink,” Personal, Indoor and Mobile Radio Communications (PIMRC), Sept. 2008.

[18] Kim, D.; Kim, J.; Kim, H.; Kim, K.; Han, Y., “An Efficient Scheduler for Uplink Single Carrier,” Personal, Indoor and Mobile Radio Communications (PIMRC), Sept. 2010.

[19] Liu Fang; She Xiaoming; Chen Lan; Otsuka Hiroyuki, “Improved Recursive Maximum Expansion Scheduling Algorithms for Uplink Single Carrier,” Vehicular Technology Conference (VTC) Spring, May 2010.

[20] Xiaoqiu Wang; Konishi, S., “Optimization Formulation of Packet Scheduling Problem in LTE Uplink,” Vehicular Technology Conference (VTC) Spring, May 2010.

[21] 3GPP TS 36.213 V8.5.0, “Physical Layer Procedures (Release 8),” http://www.3gpp.org/ftp/Specs/html-info/36213.htm, Dec. 2008.

[22] Ching-Hsu Chang; Hsi-Lu Chao; Chia-Lung Liu, “Sum Throughput-Improved Resource Allocation for LTE Uplink Transmission,” Vehicular Technology Conference (VTC) Fall, Sept. 2011.

[23] X. Qiu; K. Chawla, "On the Performance of Adaptive Modulation in Cellular Systems," in IEEE Transactions on Communications. vol. 47, pp. 884-895, 1999.

[24] Sandrasegaran Kumbesan; Mohd Ramli Huda Adibah; Basukala Riyaj, "Delay-Prioritized Scheduling (DPS) for Real Time Traffic in 3GPP LTE System," Wireless Communications and Networking Conference (WCNC), April 2010.

[25] Hongkun Yang; Fengyuan Ren; Chuang Lin; Jiao Zhang, “Frequency-Domain Packet Scheduling for 3GPP LTE Uplink,” INFOCOM 2010 Proceedings, March 2010.