Chapter One
Introduction

1.1 Preface

The Orthogonal Frequency Division Multiple Access (OFDMA) has attracted wide attention among researcher community as a most emerging candidate for multiple accesses in 4G mobile communication systems[1]. The ever increasing growth of Internet, multimedia and broadband services demands such as highly efficient multiple access techniques. Many application demands more bandwidth either in forward channel or in reverse channel, and popular applications like Internet access are bias towards downlink. So that modern multi-cellular systems feature increasingly dense base station deployments in an effort to provide higher network capacity as user traffic, especially data traffic, increases. Because of the soon ubiquitous use of Orthogonal Frequency Division Multiple Access (OFDMA) to accommodate many users in the same channel at the same time [2]in these networks, the Inter-cell interference users are assumed to be the primary source of interference, which is especially limiting for users near the boundary of the cells[3]. Data applications at higher throughputs and spectral efficiencies has driven the need to develop Orthogonal Frequency Division Multiplexing (OFDM) based 4th generation (4G) networks[4].

In OFDMA systems Inter-cell interference (ICI) is a real problem which caused by the collision between resource blocks. The overall system performance is determined by the collision probabilities and the impact of a given collision on the Signal to Interference and Noise Ratio (SINR) associated with the colliding resource blocks. [5] The Inter-cell interference coordination (ICIC) is a strategy to improve the performance of the network by having each cell allocate its resources such that interference experienced in the network is minimized, today must **c**ellular architectures must implement some form of interference control to prevent performance degradation among users in neighboring cells[6]**.**

The MSs near the cell-centre not only experience high signal quality due to their close proximity to the serving BS, but are also shielded from other-cell interference due to physical separation. Thus, it is clear that while cell-centre users do not necessitate excessive interference protection, cell-edge MSs are still highly vulnerable to CCI. Through this, Fractional Frequency Reuse (FFR) is born. In FFR, all cell centres in the system employ a frequency reuse of one, where as the cell-edges in a cluster still employ classical frequency reuse [4][5].

1.2 Problem Statement

Cellular mobile communication systems suffer from class of interference, namely **inter-cell interference**. As the frequency reuse increases, so does the **inter-cell interference** caused by other users using the same channels**,** However, **OFDMA** systems Inter-Cell Interference (ICI) still poses a real challenge that limits the system performance, Therefore, interference becomes a decisive factor that limits the system capacity, and hence, the suppression of such interference becomes of a particular importance to the design of next generation cellular networks.

1.3 Proposed Solution

A common ICIC technique is interference avoidance in which the allocation of the various system resources (e.g., time, frequency, and power) to users is controlled to ensure that the ICI remains within acceptable limits. Therefore, this thesis evaluates the use of Fractional Frequency Reuse (FFR) in order to mitigate the inter-cell interference.

1.4 Objectives

Inter-cell interference coordination (ICIC) mechanisms aim at reducing the collision probabilities and at mitigating the SINR degradation that such collisions may cause in order to improve the system performance and increase the overall bit rates of the cell and its cell edge users.

ICIC has been investigated as an approach to alleviate the impact of interference and improve performance in OFDMA-based systems.

To simulate a program for both FFR and traditional reuse schemes in order to compare between their performance in term and capabilities(probability of coverage , probability of acceptance rate and frequency reuse factor) concerning SINR and data rate. Moreover investigate the cell coverage and enhance cell edge to control and reduce the Inter Cell Interference ICI.

1.6 Research Methodology

The research has gone through two phases :

Phase one : general data collected about the research in order to implement imp Fraction Frequency Reuse.

Phase two: equations and relationship between system parameters and performance metrics simulated on MATLAB environment is used to get the results.

1.7 Thesis Outlines

This thesis composed of five chapters their details are following:

Chapter One: Introduction This chapter outlines the motivation and scope of the work.

Chapter Two: Literature Review This chapter presents some basic background on mobile broadband technologies evolution, Cellular Network Planning, resource allocation in cellular networks and we investigate frequency planning, its benefits, and research challenges.

Chapter Three: Methodology Frequency Partitioning/Reuse Approaches, Equations, Parameters used and System Model.

Chapter Four: Result and Discussion this chapter provides results from a performance evaluation of FFR and discussion about the results.

Chapter Five: Conclusion and Recommendations the main ideas presented in the thesis are collected and summarized in this chapter and recommendation for future work.

Chapter Two

Literature Review

Cellular systems are now nearly universally deployed and are under ever-increasing pressure to increase the volume of data they can deliver to consumers[7]. The last two decades have witnessed a booming in the use of cellular communication technologies. Billions of people are now enjoying the benefits of mobile communications. The problem types considered include coverage planning, power optimization, and channel assignment[8].

Coverage planning is a classical problem in cellular network deployment. A minimum-power covering problem with overlap constraints between cell pairs is considered. The objective is to minimize the total power consumption for coverage, while maintaining a necessary level of overlap to facilitate handover[9].

For High Speed Downlink Packet Access (HSDPA) networks, transmission power is a crucial factor to performance. Minimizing the power allocated for coverage enables significant power saving that can be used for HSDPA data transmission, thus enhancing the HSDPA performance. To explore this potential power saving, a mathematical model targeting cell-edge HSDPA performance has been developed. In determining the optimal coverage pattern for maximizing power saving, the model also allows for controlling the degree of soft handover for Universal Mobile Telecommunications System (UMTS) Release 99 services. In addition to the mathematical model, heuristic algorithms based on local search and repeated local search are developed[4].

For Orthogonal Frequency Division Multiple Access (OFDMA), which is used in Long Term Evolution (LTE) networks, inter-cell interference control is a key performance engineering issue. The aspect is of particular importance to cell-edge throughput. Frequency reuse schemes for mitigating inter-cell interference at cell-edge areas have received an increasing amount of research attention. In this thesis, a generalization of the standard FFR scheme is introduced. The generalization addresses OFDMA networks with irregular cell layout. Optimization algorithms using local search have been proposed to find the frequency reuse pattern of generalized FFR that maximizes the cell-edge area performance[10].

For the problems considered in the thesis, computational experiments of the optimization models and algorithms using data sets representing realistic planning scenarios have been carried out. The experimental results demonstrate the effectiveness of the proposed solution approaches[8].

2.1 Growth of Wireless Technology

Since the introduction of mobile technologies over two decades ago, wireless communication has penetrated to almost all people in the developed world, and is tremendously increasing in developing countries, thus making the mobile phone the most widespread information and communication technology (ICT) to date. Furthermore, due to the continuous growth and expansion of mobile technologies and applications, the requested traffic is increasing at a tremendous rate.

This is evident in Figure 2.1, where it is estimated that the global demand for mobile communications will increase almost six-fold within the next three years. Clearly, this is a challenging rate for mobile operators to maintain.

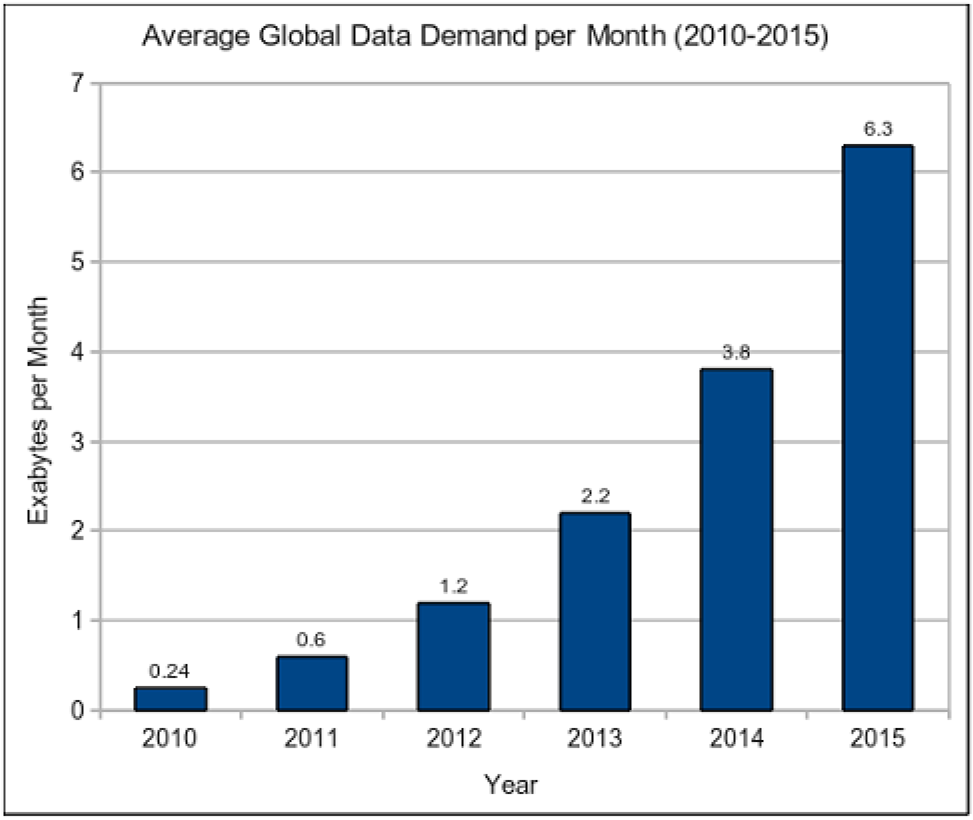


Figure 2.1 Increasing in demand of mobile services over the next years [14]

This trend is further supported by the graphs shown in Figure 2.1 where it is evident that from its humble beginnings as a worldwide service (Global System for Mobile Communications (GSM)) approximately thirty years ago, in 2011 over 85% of the Earth's inhabitants made use of mobile devices for communication .This is even more astonishing when considering that at the turn of the millennium, only approximately 15% possessed mobile phone subscriptions, and this almost solely in the developed world. Furthermore, just eleven years ago did the number of fixed-telephone line subscriptions (which now, as can be seen in Figure 2.2 has been on the decline since 2005) still exceed that of mobile devices, indicating mobile technology is quickly (or, has succeeded in) replacing landlines as the principal tool for communication. Figure 2.2 shows Global development of ICT in terms of percentage penetration for: mobile subscriptions, Internet access, users of internet, fixed-line subscriptions, mobile broadband subscriptions, and wired-broadband subscriptions.

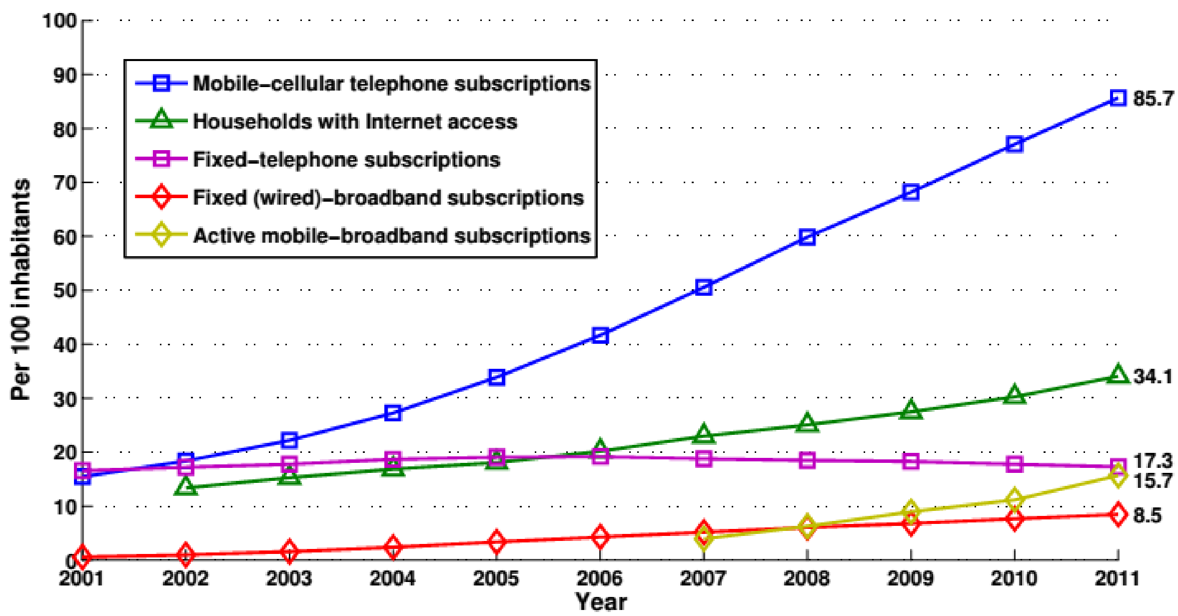


Figure 2.2 Global development of ICT [14]

Moreover, it is evident that the number of mobile broadband subscriptions has received significant uptake in the developed world in recent years which further contributing to the growing traffic demands on mobile networks.

Numerous studies indicate that the availability of Internet services can be instrumental for the development of a nation. However, fixed broadband access is often scarcely, or not at all, available in developing nations. In 2011, only 24% of the population of developing countries had access to the Internet, compared with less than 10% before 2007 .Clearly, establishing high-speed Internet access over the growing mobile networks in such countries is of paramount importance. In this light, femto-cells offer a unique opportunity to provide not only indoor coverage in developed nations, but further expand coverage in developing countries for greater wireless and Internet penetration. However, the introduction of femto-cells into existing architectures may severely complicate the operation of a network. Therefore, the development of more advanced ICIC techniques for not only established but also growing wireless networks is essential to their successful employment. Moreover, in this time of heightening environmental concerns, practical ICIC approaches which limit the complexity and energy consumption of a network are elementary in the implementation of wireless technologies[14].

2.2 Evolution of Cellular Networks

The concept of cellular communications was introduced by Bell Laboratories in 1947 to increase the communication capacity of mobile phones. In the 1970s, commercial mobile communications based on cellular technology began to appear all over the world. Systems deployed in this period are referred to as the first generation (1G) cellular networks. Nippon Telegraph And Telephone (NTT) launched the first commercial cellular network in the metropolitan area of Tokyo in 1979. Soon after, commercial cellular systems became available in Europe and America[11].

Representative systems include the Nordic Mobile Telephone system (NMT) in Scandinavia, the total access communication system (TACS) in UK, and the advanced mobile phone service system (AMPS) in America. Being analog based, 1G network suffered from many limitations such as low frequency utilization, limited service, poor communication quality, high equipment cost, and low security.

To overcome the limitations and meet the demand of mobile communications, the second generation (2G) cellular systems were developed. 2G cellular systems adopted digital transmission that is fundamentally different from the 1G cellular technology. Besides frequency division multiple access (FDMA), which is also used in 1G networks, 2G networks use more advanced access technologies such as time division multiple access (TDMA) and code division multiple access (CDMA) , as well as hierarchical cell structure[12].

Representative 2G standards include the global system for mobile communications (GSM), interim standard 136 (IS-136, aka Digital AMPS, or D-AMPS), IS-95 (aka CDMAone), and the personal digital cellular (PDC) system. Both GSM and D-AMPS were based on TDMA. GSM was firstly introduced in Europe to provide a single and unified 2G standard in European countries. Later, GSM was implemented in many countries in the rest of the world. D-AMPS evolved from AMPS and was used in America, Israel and some of the Asian countries. IS-95 systems were based on CDMA, which allowed users to simultaneously access as low-cost, intermediate solutions for data services, while developing the third generation (3G) systems. The intermediate steps in the transition from 2G to 3G were called 2.5G and 2.75G. One such step was high-speed circuit switched data (HSCSD) for GSM. HSCSD was easy to implement and required low cost in deployment. However, since it was still based on circuit switching, it suffered from inefficient frequency utilization. Another technology was the general packet radio service (GPRS) . Using packet switching, GPRS had better frequency utilization in handling best-effort data communication. GPRS utilizes dynamically free TDMA channels in GSM systems, and can provide a speed of up to 114 kbps. Later, with the introduction of 8PSK encoding, GPRS evolved to enhanced data rate for GSM evolution (EDGE) , aka enhanced GPRS. EDGE increased the data rate to 384kbps and was backward-compatible. At the same time, within the CDMA family of standards, IS-95 evolved to the CDMA2000 1 times radio transmission technology (CDMA2000 1xRTT), which supported a data rate of 153kpbs[13].

Soon after the commercialization of 2G networks, development of 3G cellular communication systems started. The goal of 3G was to provide high data rate and multimedia data services for mobile Internet. Three major families of specifications, namely CDMA2000, wideband CDMA (WCDMA), and time division synchronous CDMA (TD-SCDMA) were accepted as 3G standards by the international telecommunications union (ITU). CDMA2000 was developed from IS-95 and mostly used in North America and South Korea. WCDMA, aka universal mobile telecommunication systems (UMTS) was developed from GSM, with efforts in Europe and Japan. Since GSM dominated the 2G commercial markets, the evolution path GSM-GPRS-EDGE-WCDMA was very natural. At present, WCDMA is the most widely implemented 3G technology. TD-SCDMA was China's 3G standard, proposed by Datang Telecom and was mostly used in China.

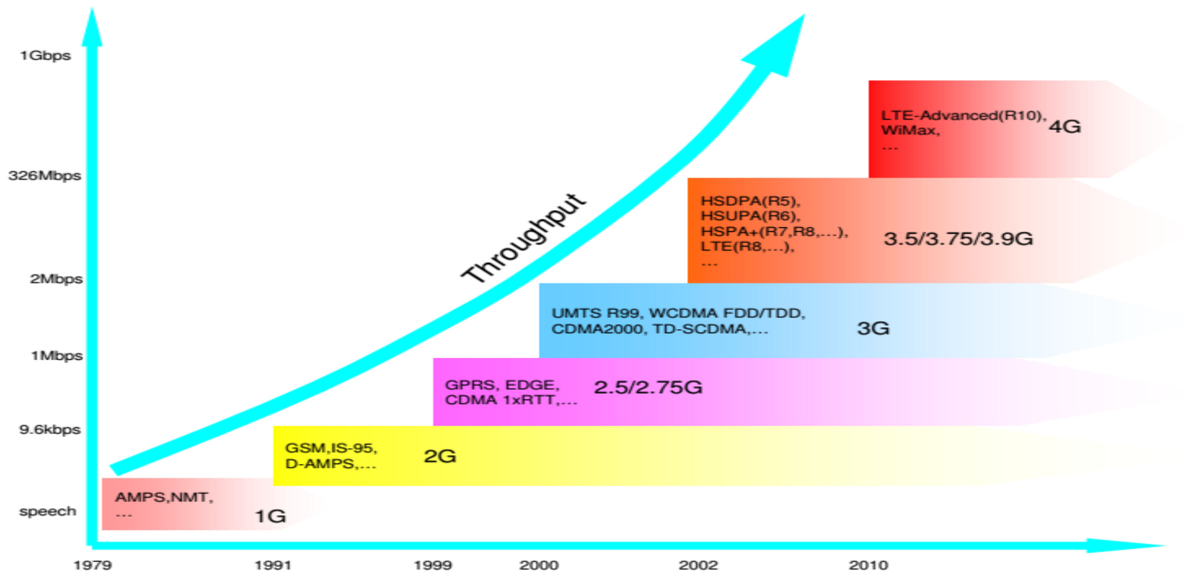
Since the first release by 3rd generation partnership project (3GPP), referred to as release 99 (R99), WCDMA has evolved rapidly. In 2002, Release 5 (R5) was finalized to include high speed downlink packet access (HSDPA). In 2005, Release 6 (R6) became available, supporting high speed data services on the uplink by introducing enhanced uplink (EUL), aka high speed uplink packet access (HSUPA). In 2007, Release 7 (R7) introduced evolved high speed packet access, aka HSPA+, to support higher order modulation 64 QAM and Multiple Input and Multiple Output (MIMO) technology. In Release 8 (R8), which was finalized in 2008, developments of two standards were specified, HSPA+ and long term evolution (LTE). HSPA+ was enhanced by a dual-carrier HSPA (DC-HSPA) which doubled the throughput by combining two WCDMA radio channels. HSPA+ was a backward-compatible standard, allowing operators that have heavily invested in WCDMA to offer new features to their subscribers with legacy UMTS terminals. At the same time, the first release of LTE was specified in R8. LTE represented a new evolution path. The LTE standards adopted orthogonal frequency division multiple access (OFDMA) for downlink and single carrier FDMA (SC-FDMA) for uplink, to provide higher peak data rate and flexible bandwidth usage. Additional features, such as femto-cell, were introduced in Release 9 (R9) in 2009.

The ultimate goal of LTE is to pave the way for the fourth generation (4G) systems. Peak data rate of LTE targets 100 Mbps at downlink and 50 Mbps at uplink with a 20 MHz bandwidth and 2×1 MIMO antenna. In comparison to R6, average user throughput of LTE is expected to be 3-4 times higher at downlink, and 2-3 times higher at uplink. In LTE, spectrum utilization is scalable over blocks of 5, 10, 15, and 20 MHz. Blocks smaller than 5 MHz are also supported. As LTE is not supposed to be backward-compatible, it enjoys the freedom of implementing the latest transmission technologies. Besides the new air interface, advanced antenna solutions such as MIMO, beam-forming are also utilized. A flat all IP core network, namely system architecture evolution (SAE) , is designed to replace the GPRS core network and to support high throughput with low latency in the radio access network (RAN). Despite the economic crisis in 2008 and 2009, trials of LTE networks have not been interrupted. The first commercial LTE service was provided by TeliaSonera in Stockholm and Oslo in 2009. The network delivers a downlink speed between 20 Mbps and 80 Mbps.[15]

To meet the future demand of mobile broadband, the research on LTE-advanced (LTE-A), which represents a 4G standard, is also underway. LTE-A specifications will be part of Release 10 (R10). LTE-A will include more advanced technologies like clustering SC-FDMA, relaying, coordinated multi-pointing transmission/ reception (CoMP), higher order MIMO transmission, etc., to take the peak data rate up to 1 Gbps for low mobility scenarios. [15]

The above discussion applies to the development of standards in 3GPP as it is given in Figure 2.3. Besides LTE, worldwide interoperability for microwave access

(WiMAX), specified in IEEE 802.16 series of standards, is also evolving towards fulfilling the requirement of 4G systems,



gure 2.3 Evolution of cellular technology [15]

2.3 Cellular Network Planning and Optimization

Network planning and optimization play a key role in reducing the capital expenditure (CAPEX) and operational expenditure (OPEX) for deploying and expanding cellular systems. Typically, radio network planning begins with a definition and dimensioning stage, which includes traffic estimation, service definition, coverage and capacity requirements, etc. It is traditionally considered as a static process. Some of the main tasks are site selection for base station location, location area, routing area planning, and radio resource management (RRM) strategies. Besides fulfilling the initial requirements such as coverage and capacity, radio resource has to be acquired in a way that the cost is minimized. Optimization is a long-term process before and after the launch of a network. The process applies various methods to maximize the system performance by optimally configuring the network and utilizing its resources. Traditionally, a large amount of manual tuning has been used in the optimization process. Nowadays, advanced optimization tools have been developed to automatically optimize the parameters for maximizing system performance, making the optimization process much more efficient[15].

Analog-based 1G cellular network did not pose much requirement to planning and optimization. Having low capacity requirement, the key problem is to provide a satisfied degree of coverage. With the success of 2G networks and the increasing amount of voice and data demand, cellular systems have become more and more complex. New radio access technologies also introduce new challenges to the planning and optimization processes. In the following, we discuss some main planning and optimization problems in cellular networks. More detailed discussions can be found in, for example, a very fundamental planning task in cellular networks is the base station (BS) location selection. To deal with the increasing user demand, more and more base stations are needed to provide satisfactory services. The best outcome of optimization problem is to determine how many and where base stations should be located in order to meet coverage and capacity requirements. An illustration of the problem is shown in Figure 2.4 the red spots require BS installed while the blue spots do not.

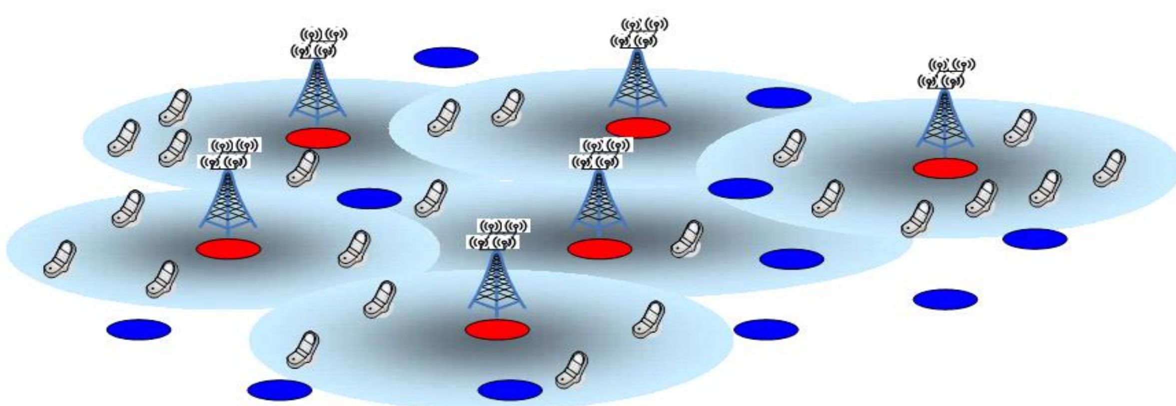


Figure 2.4 An illustration of base station location [15].

In general, the base station location problem is NP-hard [21] in problem complexity. For 2G networks, to a large extend, the task can be regarded as a type of set covering problem, the performance is mainly coverage-driven and mostly depends on the propagation model. In this case, BSs are located among the candidate locations so that signal strength is high enough for the areas to be covered. For 3G networks with WCDMA, this is however not enough as the single-to-interference ratio (SIR) needs to be taken into account. Besides the location of BS, coverage itself is heavily influenced by the traffic distribution[22].

Thus, BS location in UMTS networks has to consider power control which in its turn is tightly connected to traffic distribution. In the past years, BS location has been extensively studied for both 2G and 3G networks. Both mathematical modeling and heuristic algorithms have been proposed for problem solution.

2.3.1 Frequency Planning

Along with BS location, another vital issue in 2G GSM networks is to deal with the frequency assignment problem (FAP). To avoid interference in GSM networks, neighboring cells should use different frequencies; see Fig. 3 for an illustration.

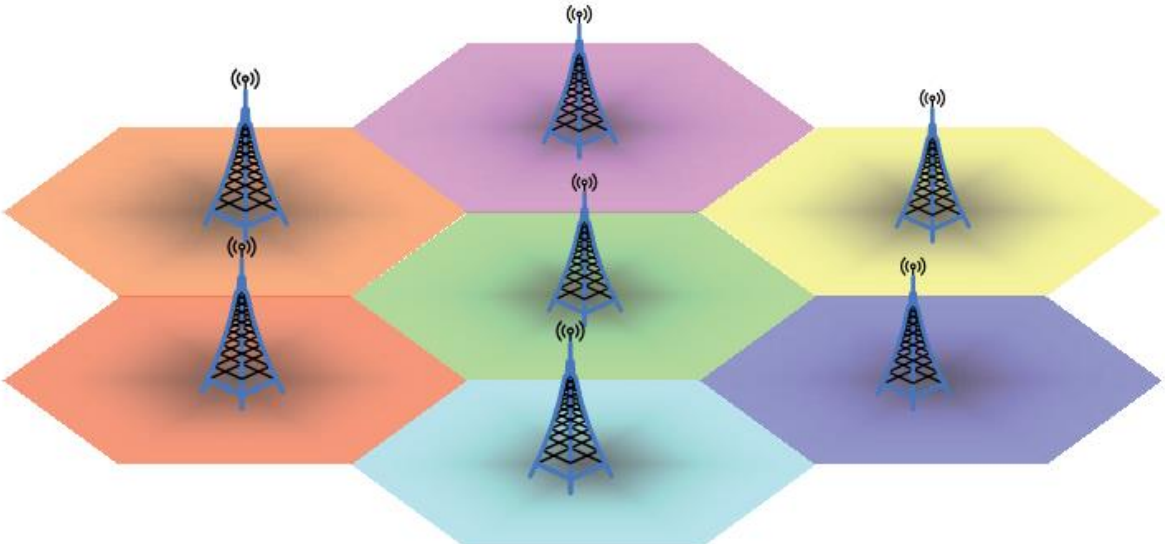


Figure 2.5 An illustration of frequency assignment [15]

Due to the high site density and the scarcity of the spectrum resource, frequency allocation has to be carefully planned so that high spectral efficiency and low interference can be achieved. Many optimization methods and algorithms have been proposed for FAP in GSM networks. Planning strategies for more advanced frequency use, such as frequency hopping (FH) and dynamic channel allocation (DCA) have also been proposed.

In WCDMA networks, the frequency reuse factor is one, so FAP is not present. For the new radio interface OFDMA adopted in LTE networks, frequency optimization becomes again a key issue. LTE is designed to scale well in the spectrum availability. The spectrum is divided into a large number of subcarriers which are orthogonal to each other. This enables a higher flexibility in respect to resource allocation. With OFDMA, intra-cell interference does not exist because of the orthogonality, but inter-cell interference may become the performance-limiting factor, especially for cell-edge users with poor radio signal condition. Sub-carrier allocation has to be done carefully to mitigate intercell interference. To this end, fractional frequency reuse (FFR) and soft frequency reuse (SFR) schemes have been proposed. With the evolution from 3G to 4G networks, inter-cell interference mitigation is attracting more and more research attention.

2.3.2Adaptive resource allocation in cellular networks

A wireless cellular network comprises base stations serving users. The assignment of users to the base stations depends on the strength of receiving signal. As a mobile device can usually receive signals from several base stations, it is typically assigned to the base station with the strongest received signal. Signals from other base stations are known as intercell interference which may cause a low Signal to Interference plus Noise Ratio (SINR). Consequently this decreases the transmission data rate of the users.

In order to avoid excessive intercell interference traditional cellular networks employ a fixed frequency reuse pattern so that neighboring base stations do not share the same frequency.

2.3.3 Network Optimization

In this part we consider multiple base station cooperative networks where an OFDMA scheme is employed within each cell, and no two links in each cell can use the same subcarrier at the same time. So, there is no intracell interference. Given a fixed frequency and power allocation for all transmitters, the network optimization problem is that of coordinating the subcarrier assignment.

Neighboring cells do not interfere with each other. The traditional fixed frequency reuse schemes are effective in minimizing intercell interference, but are also resource intensive in the sense that each cell requires a substantial amount of nonoverlapping bandwidth, so that only a fraction of the total bandwidth can be made available for each cell. Consequently, the standardization processes for future wireless systems have increasingly targeted at maximal frequency reuse, where all cells use the same frequency everywhere.

Wireless channels are fundamentally impaired by fading, propagation loss, and interference. Two types of cooperative network that specifically address the issues of intercell interference and path-loss:

• Base station cooperation: While in traditional networks base stations were operating independently, this type of cooperative network explores the possibility of coordinating multiple base stations. In these networks the transmission strategies among the multiple BS are designed jointly. In particular, the base stations may cooperate in their power, frequency, and rate allocations in order to jointly mitigate the effect of intercell interference for users at the cell edge.

• Relay cooperation: This type of cooperative network explores the use of relays to aid the direct communication between the base station and the remote subscribers in order to combat against the path-loss characteristic of wireless channel.

In both types of cooperative networks, resource allocation is expected to be a crucial issue.

In matter of fact a wide range of techniques and schemes is presented in order to improve the throughput of the cell-edge users by reducing or suppressing the ICI some important techniques or schemes for ICI shown in figure 2.6.

2.3.3.1The techniques and schemes

-The main ICI mitigation techniques include :

-(1) Interference randomization, where some cell-specific scrambling, interleaving, or frequency-hopping (spread spectrum) .

-(2) interference cancelation: where the interference signals are detected and subtracted from the desired received signal, or if multiple antenna system is employed, the receiver can select the best quality signal among the various received signals.

-(3) adaptive beamforming: where the antenna can dynamically change its radiation pattern depending on the interference levels.

Interference avoidance schemes represent the frequency reuse planning which are shown below used by the network elements to restrict or allocate certain resources (in both frequency and time domains) and power levels among users in different cells.

-Interference Avoidance Schemes include:

-(1) Traditional Frequency Reuse : it is simplest scheme to allocate frequencies in a cellular network by using reuse factor of 1which leads to high peak data rates. However, in this case, higher interference is observed on cell edges. The classical interference management is done by using reuse ratio 3 by using this reuse factor interference is low but there will be large capacity loss because only one third of resources are used in each cell.

-(2)Fractional Frequency: inter-Cell Interference Avoidance Schemes To avoid the shortcomings of the conventional frequency reuse schemes, the fractional frequency reuse (FFR) scheme is introduced to achieve a FRF between 1 and 3. FFR divides the whole available resources into two subsets or groups, namely, the major group and the minor group. The former is used to serve the cell-edge users, while the latter is used to cover the cell-center users.

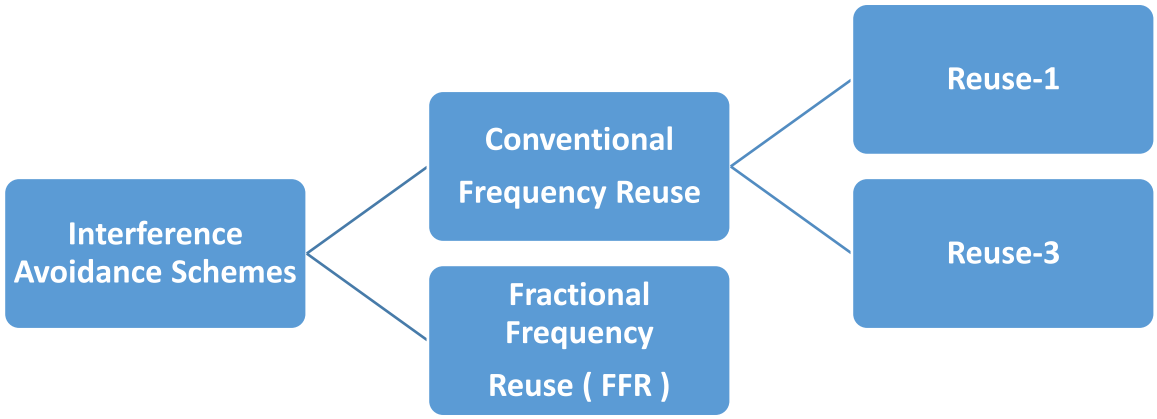


Figure 2.6 Interference Avoidance Schemes

Chapter Three

Performance Analysis of Fractional Frequency Reuse

The issue of inter-cell interference coordination (ICIC) for high-capacity next generation wireless networks is addressed in this thesis. With the ever-increasing uptake of mobile users, wireless communication has evolved into a utilities similar to water and electricity, needed by almost all people of today's modern society. Furthermore, the large demands for multimedia services such as Internet and TV are fast rendering state-of-the-art cellular systems incapable of supporting the requested traffic in the network. Thus, smaller cell sizes, micro-cells and full frequency reuse are implemented to increase the spatial reuse of wireless resources over a geographical area. However, while this inherently increases the signal-to-noise ratio (SNR) in the network, the high interference caused by the dense cellular structure harms the achievable spectral efficiency. Furthermore, the additional power consumption from the multitude of base stations (BSs) indicates the necessity for more energy efficient ICIC techniques for modern networks[16].

On the other hand, recent studies indicate that a substantial portion of wireless traffic originates indoors. Poor signal reception through walls severely inhibits the operation of indoor data services, attracting considerable interest in the concept of femto-cells. Thus, femtoBSs (FBSs), which are low-cost, low-power, short range, plug-and-play BSs, aim to enhance indoor coverage, alleviating this burden from the macro-cell sites. Furthermore, macro-cell coverage is extended through pico-cells, which provide micro-cells within a macro-cell in order to further augment the availability of wireless resources. Hence, future wireless networks are moving towards heterogeneous architectures, with multiple access points (APs) available in each macro-cell .

This work focuses primarily on the development of novel ICIC techniques to manage the upcoming challenges of future heterogeneous networks (HetNets). The coordination of the dense macro-cellular environment, femto-cell deployment and additional micro-cells is addressed, with special attention paid to spectrally and energy efficient operation of these systems[17].

3.1 Frequency Partitioning/Reuse Approaches

The ICIC techniques exploit only frequency domain through partial use of radio spectrum, i.e., frequency partitioning, and/or transmit power domain through link power adaptation. Whereas, the Enhanced Inter Cell Interference Coordination (eICIC) scheme along with the advanced interference cancelation (IC) capabilities in the terminal receivers enables operators to deploy low power small cells under the coverage of high-power macro cells using the same channel[4].

The typical frequency partitioning methods used for ICIC and eICIC techniques are briefly explained in the following:

3.1.1 Full Frequency Reuse: There is no frequency partitioning, i.e. Frequency reuse factor (FRF) is 1, among the macro BSs of the same network, and each macro BS transmits with uniform power using the entire system bandwidth substantially creating inter-cell interferences at the cell edges both in the downlinks and in the up links. The 4G small cells such as femto cells, have been provisioned with this approach under the name Cochannel Deployment, wherein the same spectrum is reused simultaneously among the radio frequency (RF) entities of macro cell and femto cells in the same geographical area.

If interference is ignored, then it is clear that the capacity of mobile cellular network would be maximised when all cells share the entire system bandwidth; this is called full frequency reuse, and will be utilised in LTE deployments [23]. In light with the previous figures, such channel

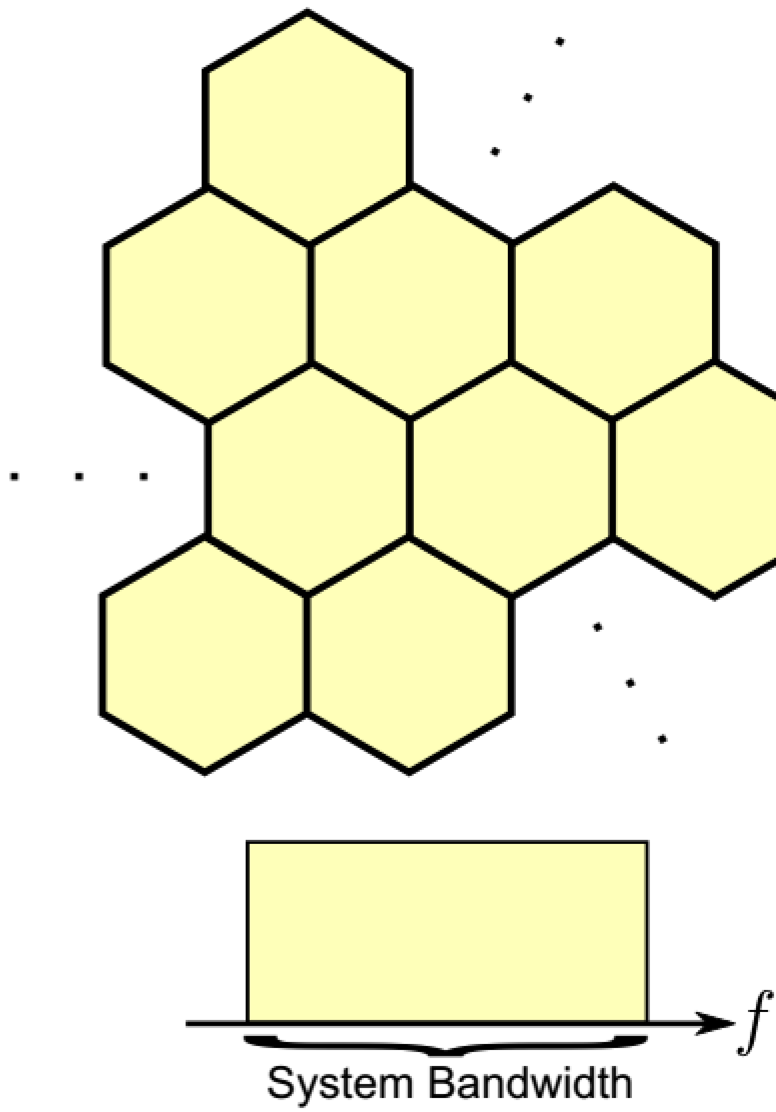


Figure 3.1: Full frequency reuse [25]

Allocation is depicted in Figure 3.1, where it is clear that all BSs can transmit on the full available bandwidth. Clearly, this results in very high CCI in all cells from their immediate neighbours. Since a further goal of LTE is acceptable cell-edge performance, it is clear that ICIC is vital to the performance of these networks. Among others, the main techniques necessary for LTE ICIC are:

- Interference mitigation/alignment.
- Power control for interference reduction.
- Frequency domain scheduling for interference coordination.
- Partial/dynamic frequency reuse at cell-edges.

To name a few, This dissertation proposes ICIC methodologies that implement a combination of the above mechanisms in order to maintain full frequency reuse, and further supply adequate performance to cell-edge users.

3.1.2. Hard Frequency Reuse: This approach, which is typically used in GSM and LTE release 8/9, the entire sub-carriers are partitioned into 3, 4 or 7 disjoint sets, i.e., with FRFs of 3, 4 or 7 respectively, and are assigned to the individual macro BSs in such a way that any adjacent macro cells pair must use disjoint, i.e., orthogonal, set of partitioned sub-carriers. This approach is the basis for cell clustering engineering. This approach maximally eliminates cell edges interference but causes decrease in the spectrum reuse efficiency by a factor equal to FRF. The 4G small cells such as femto cell technology, has been provisioned with this approach under the name Dedicated Channel Deployment, wherein femto base stations (FBSs) and the macro base station (MBS) utilize radio spectrum orthogonal to each other, and there is no spectrum re-use benefits and no co-channel interference issues.

The original mobile communications systems consisted of a high power BS that was meant to serve a large geographical area. However, this not only severely limited the number of concurrently served users, but also greatly restricted, or did not allow, for roaming of users outside this area. Therefore, a mechanism was developed to not only expand the coverage area of mobile networks, but also to increase the number of accessible channels (and hence, users); this is called the cellular concept. In this concept, rather than a single BS, many lower-power BSs, each allocated a set of radio channels, are utilized that each cover to smaller geographic area called a cell. However, it is clear that if two neighboring BSs were to utilize a similar set of radio resources, these would interfere with each other and, consequently, degrade the performance in each cell. Thus, cells were formed into clusters, where in each cluster every cell is allocated a different set of channels than any of its neighbors. Moreover, these clusters can then be repeated over much larger geographical areas, such that the system bandwidth is reused several times within the system. This orthogonal allocation of radio resources within clusters of BSs in a system is known as frequency reuse. Figure 3.2 shows the most primitive form of frequency reuse.

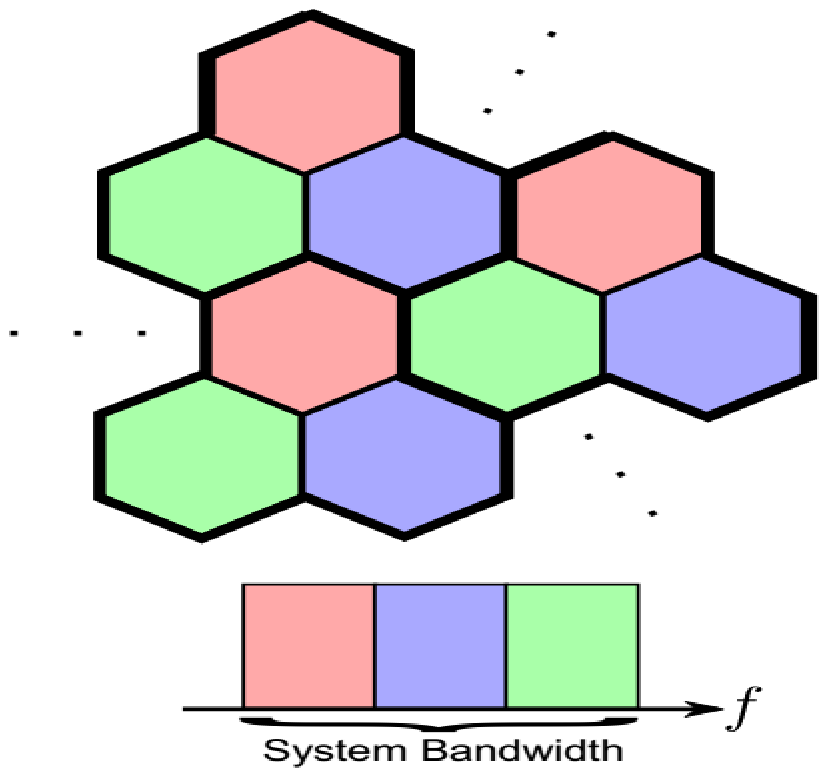


Figure 3.2: The most basic form of frequency reuse [25]

Where the available bandwidth is divided into three (can be increased for higher interference mitigation, e.g, 4, 7, 9,. . .) equally-sized orthogonal bands, and distributed such that no neighboring cells utilize the same band. Here, three clusters are shown.

Where the system bandwidth is divided into three orthogonal subbands (corresponding to a "frequency reuse of three") and each cluster is formed by three neighboring cells. Therefore, due to the geographical separation between cells hosting the same frequencies, the co-channel interference (CCI) is mitigated.

Unfortunately, there is a clear drawback to this scheme: the diminished spatial reuse of re-19 sources over the network area. Evidently, reusing more channels in each cell would increase the number of servable users. However, this reuse is limited by the interference that is tolerable in each cell, and thus cluster sizes, and hence the physical separation of channels, can be increased in order to protect co-channel cells from each others interference. Therefore, there exists a clear tradeoff between bandwidth utilization and interference limitation.

3.1.3 Fractional Frequency Reuse: With this approach the system bandwidth is divided into two parts. One part is used through Full Frequency Reuse method, typically for the central cell UEs, and the second part is used through Hard Frequency Reuse method, typically for the cell edge UEs. Therefore, this approach combines the benefits of the first two methods while avoiding their drawbacks, and is useful in the uplink scenario where cell-edge UEs experience severe intercell interference. The 4G small cells such as femto cell technology, has been provisioned with this approach under the name Partial Cochannel Deployment, wherein some parts of radio spectrum utilized by the FBSs are orthogonal to that of the MBS, while other parts of radio spectrum are shared among FBSs and MBS.

In conjunction with growing demands for mobile services, the necessity arose for enhanced system capacity. According to Shannon [25], the most effective method for improving capacity is increasing the available bandwidth. Without adding to the already expensive system bandwidth, this is performed by improving the spatial reuse of resources, and can be easily achieved given two realizations:

- MSs near the cell-centre not only experience high signal quality due to their close proximity to the serving BS, but are also shielded from other-cell interference due to physical separation.

- On the other hand, it is clear that cell-edge users will receive stronger interference from other cells, simply due to proximity. Furthermore, these MSs will experience degraded performance due to the large distance to their BS.

Thus, it is clear that while cell-centre users do not necessitate excessive interference protection, cell-edge MSs are still highly vulnerable to CCI. Through this, FFR is born. In FFR, all cell centres in the system employ a frequency reuse of one, where as the cell-edges in a cluster still employ classical frequency reuse . This is portrayed in Fig. 2.8. When comparing this channel allocation to that in Fig. 3.3, it is clear that the number of radio resources that are available in each cell are twice that of classical frequency reuse, hence doubling the system capacity (for higher cluster orders, this gain is even greater). Of course, additional CCI is now present in the system, somewhat degrading performance, however the gained bandwidth greatly outweighs this (minor) performance reduction.

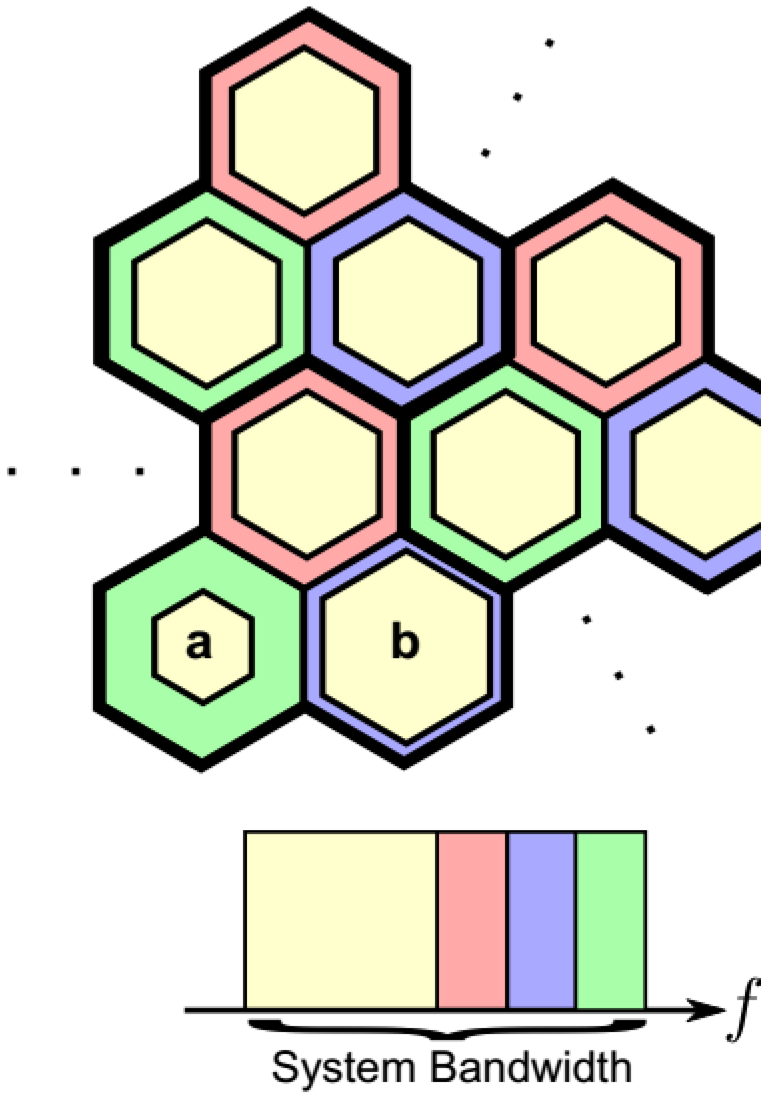


Figure 3.3 :Fraction frequency reuse [25]

Thus we can conclude that FFR improves the spatial reuse of resources by reusing the same band in the center of the cell, and protecting the edges through standard frequency reuse.

3.1.4. Soft Frequency Reuse: This is the same as Full Frequency Reuse approach but with the use of non-uniform transmit power spectrum, and is useful in the down-link.

These frequency assignment approaches adopted in the conventional cellular concept are static, i.e., these take place after careful planning as long term configurations, and these does not take into account network dynamics through active information exchange among the interfering nodes. The HetNets can exploit these approaches through dynamic configurations with the network information exchange that can be done with separate signaling interface X2 which is provisioned for eICIC functionality in each 4G HetNet node both in frequency and time domains.

One such group of techniques already proposed is that of SFR, or adaptive FFR [24]. In SFR, the full bandwidth is utilized in all cells, however transmit power control (i.e., reduction) is performed on a subset of the resources in each cell, where the full power resources are reused in a similar manner to frequency subbands in FFR. In fact, a methodology for SFR has already been standardized for LTE.

An implementation of SFR is shown in Figure 3.4, in which fixed power masks are implemented in the cell-centres where, similar to the motivation for FFR, users are in need of less power to achieve their SINR requirements (due to BS proximity). Furthermore, the diminished transmit power provides interference reduction for neighboring cells. At the cell-edges, full power transmission is allocated to enable MSs to maintain sufficient SNR. Finally, study is conducted in [25] into adaptive SFR techniques, where the power masks on the individual subbands may be tuned to the immediate interference environment. As mentioned previously, this dissertation investigates exactly such methods, by which full frequency reuse is maintained through interference mitigation via intelligent scheduling and power control.

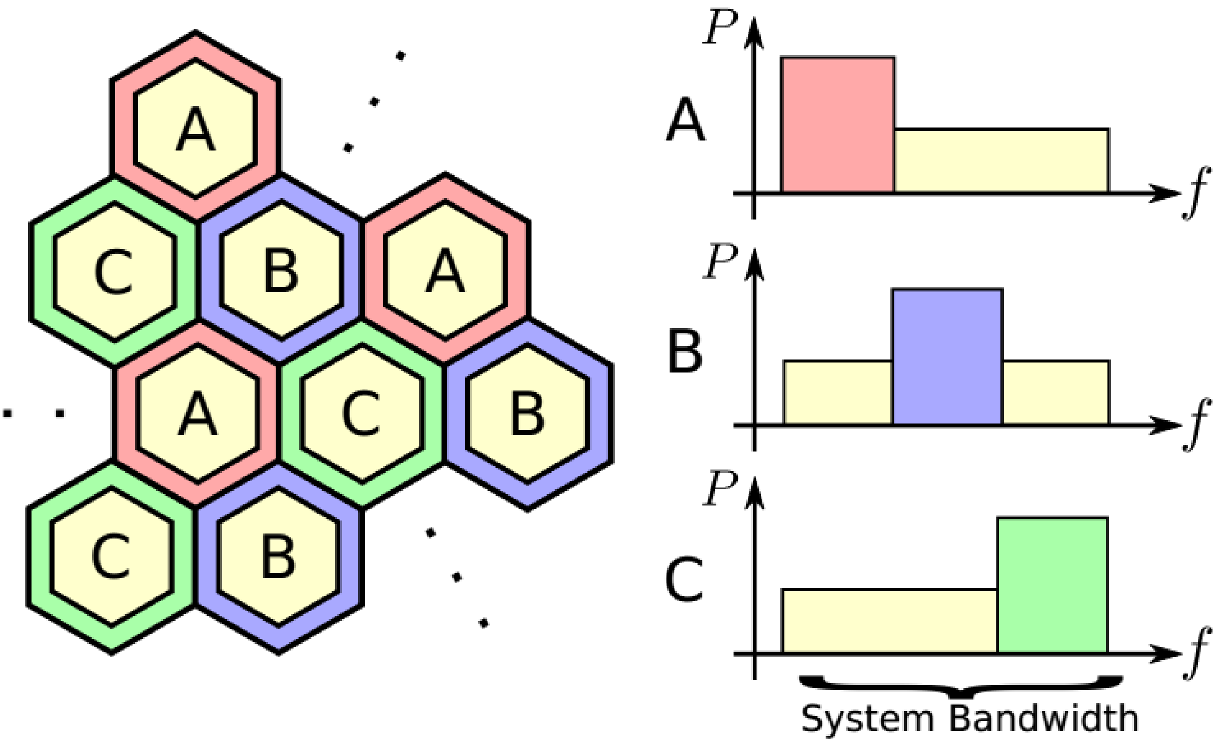


Figure 3.4 SFR optimizes the full frequency reuse of resources [25]

3.2 System Model

It's assumed that all the BSs transmit with an equal power P. The path loss exponent is given by, α and α^2 is the noise power. We assume that the small-scale fading between any BS z and the typical[1] mobile in consideration denoted by g_z, i.i.d exponentially distributed with mean g_z (corresponds to Rayleigh fading). The set of interfering base stations is Z, where base stations that use the same sub-band as user y. We denote the distance between the interfering BS z in and the mobile node in consideration y by R_z. The associated Signal to Interference plus Noise Ratio (SINR) is given in equation 3.1

$$\mathrm{SINR} = \frac{\mathrm{P g_y r^{-\alpha}}}{\sigma^2 + \mathrm{PI_r}} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (3.1)$$

where for an interfering BS set Z is given as equation 3.2

$$I_r = \sum_{z \in Z} g_z R_z a^{-\alpha} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.2)$$

In the above expression, we have assumed that the nearest BS to the mobile y is at a distance r, which is a random variable.

Where,

g is statistical distribution and is fading value or value for fading, shadowing and any other desired random effect with mean (1/μ). When g is also exponential then simpler expression will result.

h is exponential random variable(h~ exp(μ)).

r is distance from mobile to its base station.

R is distance from the mobile to other stations on same reuse assignment.

α is path loss coefficient.

σ^2 is noise power.

And 'i' represents each of the mobiles which are interfering with the mobile whose SINR is being calculated. All above are for single transmit and single receive antenna and similarly we consider that there is no same cell interference due to orthogonal multiple access (OFDMA) within a cell. The noise power is assumed to be additive and constant with value of σ 2 but no specific distribution is assumed.

The coverage probability is the probability that a typical mobile user is able to achieve some threshold SINR, i.e. it is the complementary cumulative distribution function (CCDF).

Mathematically, coverage probability is modeled by equation 3.3

$$P_c\ (T,\lambda,\alpha\ ,) \approx\ P[SINR > T\ ] \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(3.3)$$

Where, T is target threshold SINR value.

The CDF gives P [SINR≤ T] so CCDF of SINR over the entire network is probability of coverage too. The achievable rate shows $\tau \rightarrow \ln(1+ SINR)$, i.e. Shannon bound. τ has unit nats/Hz (since log is base e and 1 bit =ln(2)=0.693nats). The term Traditional Frequency reuse and Conventional Frequency reuse is used in same sense thus, somewhere it is mentioned as conventional frequency reuse which means the same. The system is simulated in MATLAB and mathematical expression basis is reference. The environment assumed is static, plane terrain, urban area with Hexagonal geometry with symmetric alignment of eNBs. This makes the simulation a bit simpler. Here in this thesis we are doing comparative analysis thus, this assumption also makes good sense for analysis though we are not assuming real time scenario.[19]

3.2.2 Parameters used:

User Equipment's intensity, $\lambda = 5$

Path loss exponent, $\alpha = 4$ (Urban Area)

Avg. SNR (to calculate noise) = 10 dB

Cell radius of 1Km

Total tiers considered are 15 and users are distributed randomly within first 9 tiers from center cell. SINR threshold to distinguish cell edge users and cell center users is 15dB. Cell radius of 1Km is taken during observation. The environment considered is totally static and flat terrain.

Simulation is carried out for number of times to calculate SINR and rate for user equipment and its mean value is taken during final plot so that best result is obtained[20].

Chapter Four

Results and Discussions

4.1 Simulation Parameters

This chapter discusses the results of the simulation after executing the system model which described in the previous chapter with an interactive Graphical User Interface (GUI) to manipulate them. Also analysis and comment on these results have been described. Moreover the simulation parameters are used in this work are given in table 4.1

Table 4.1 Simulation Parameters

Parameter	**Value**
User Equipment's intensity(λ)	5
Path loss exponent(α)	4
Avg. SNR (to calculate noise)	10 dB
Cell radius	1Km
Total tiers considered	15
SINR threshold	15dB

4.1 Signal to Interference Noise Ratio (SINR) of Traditional Reuses and FFR

Fraction Frequency reuse is always in seek of scheme which has SINR performance that matches with Traditional reuse 3 and Rate that match with Traditional reuse 1. It is clear that FFR which is shown in figure 4.3 has better performance in SINR than Traditional reuse 1 (Figure 4.1) and similar performance as of Traditional reuse 3

(Figure 4.2)and gives better coverage around 2 times more than reuse 3 in SINR of 15 to 30 dB.

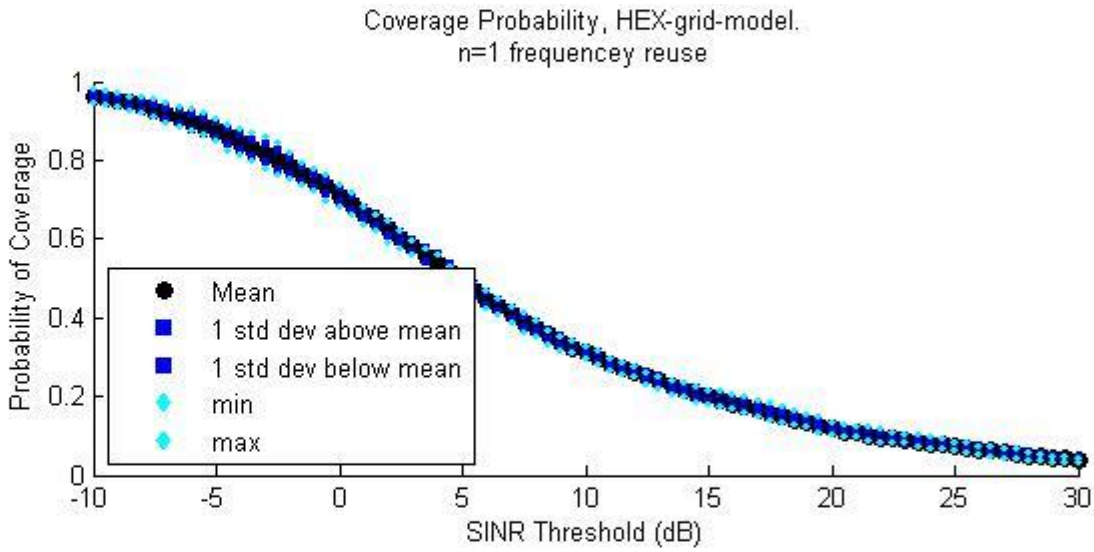


Figure 4.1 SINR of Traditional Reuse 1

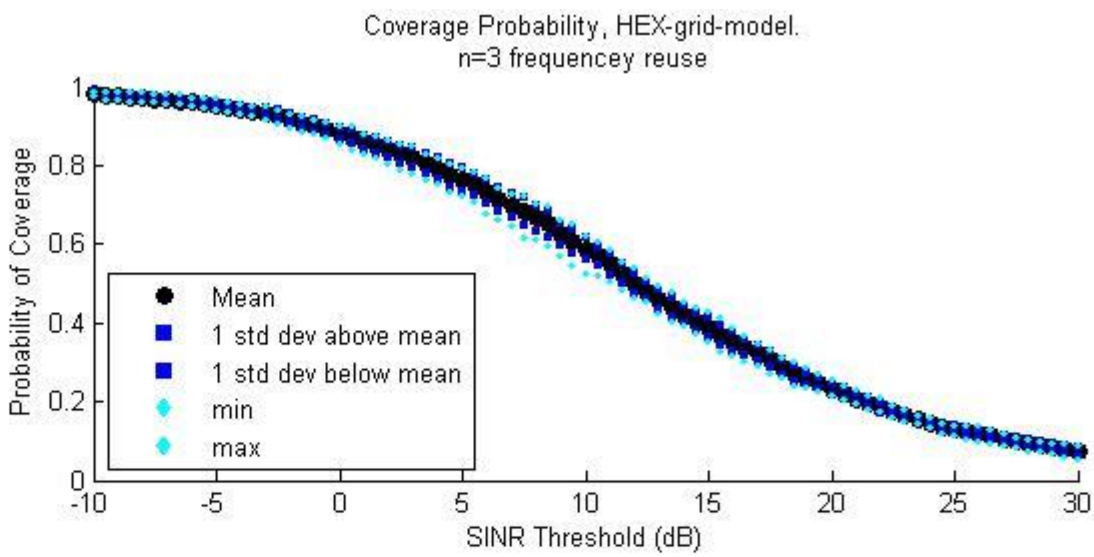


Figure4. 2 SINR of Traditional Reuse 3

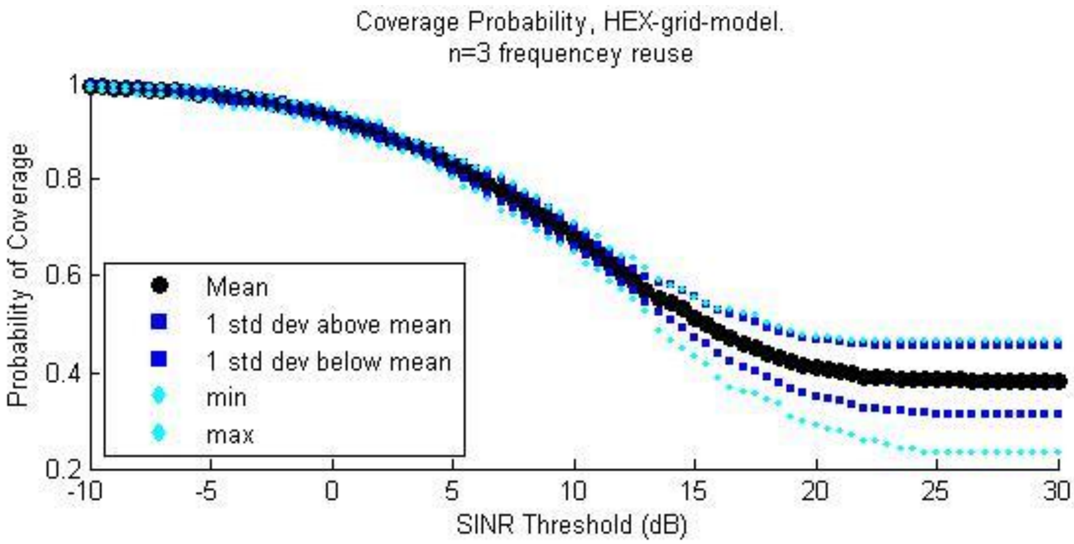


Figure 4.3 SINR of Fraction Frequency Reuse 3

4.2 Map of the model

Regarding to the simulation type, there is a map of the model is always displayed as well. In particular, figure 4.4, figure 4.5 and figure 4.6 illustrate traditional reuse 1and 3 and FFR (reuse 3) reuse deployment respectively. Each colored circle represents a base-station of a certain assignment. The small black dots represent mobiles.

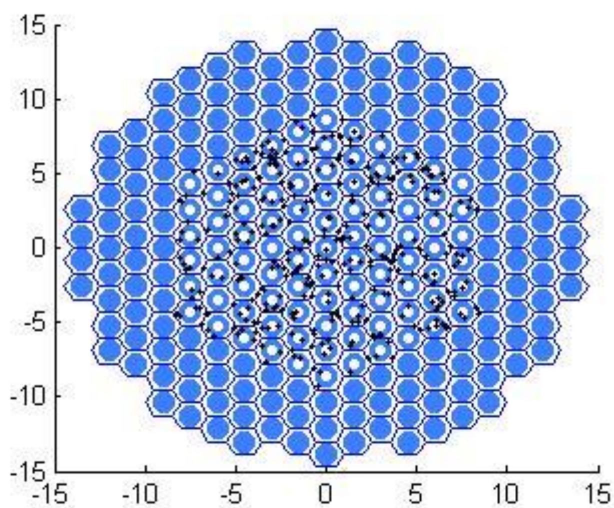


Figure 4.4 Map of Traditional Reuse 1

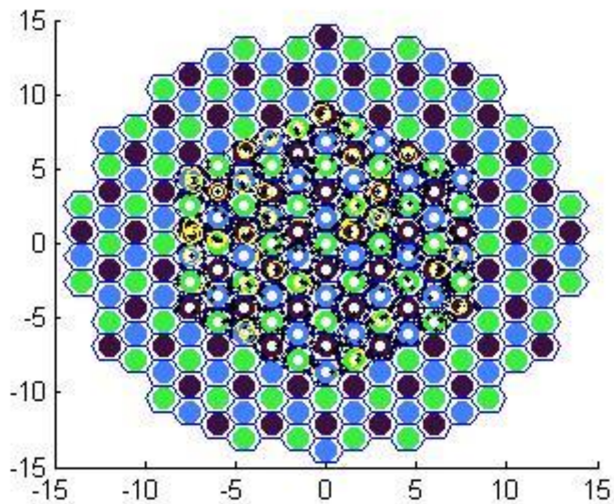


Figure 4.5 Map of Traditional Reuse 3

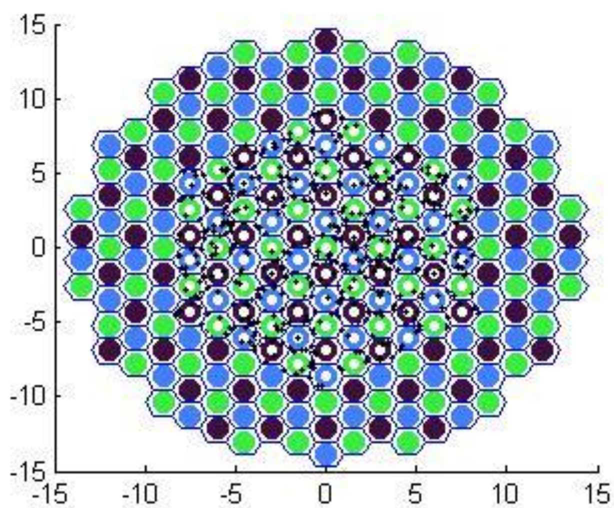


Figure 4.6 Map of Fraction Frequency Reuse 3

4.3 Data rate of Traditional Reuses and FFR

The received data rate of three reuse types is shown in Figure 4.7, Figure 4.8 before fractioning and Figure 4.9 after fractioning respectively where the Percentage gain in Probability of Acceptance rate (PAR) calculation for FFR reuse 3 with respect to Traditional Reuse 1 and Reuse3 is calculated in table 4.2.

Table 4.2 is tabulation of figure 4.7, figure 4.8 and figure 4.9. This shows that for FFR we obtain 38.5 % and 98.6 and 143.4% (value is more than doubled) at 2, 2.5, and 3 nats/Hz rate threshold values respectively relative to Traditional reuse 3. Similarly, gain of 44.9%, 97.2%, 160.2 % and 209.2 %(value is nearly tripled) are for 3.5-5 nats/Hz rate thresholds relative to Traditional reuse 1 for FFR. These observations clearly show that FFR has better performance than Traditional frequency reuse inspite of they show initial degradation but better performance after 1.5 nats/Hz in rate. Now, observing figure 4.2 and figure 4.9 FFR has SINR as Traditional reuse 3 but both figure 4.8 and figure 4.9 clearly show that FFR has better performance than Traditional frequency reuse.

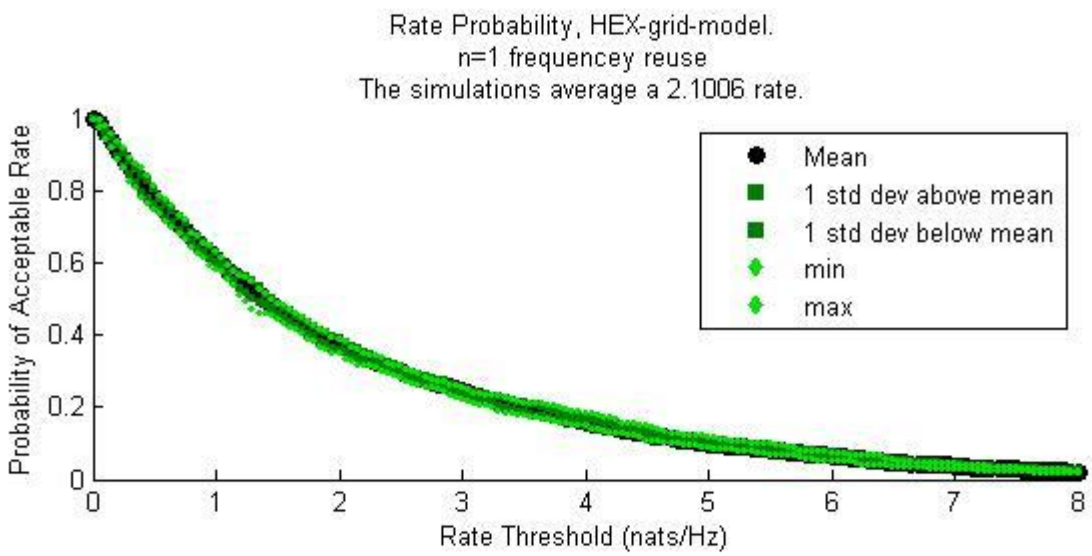


Figure 4.7 Data Rate of Traditional Reuse 1

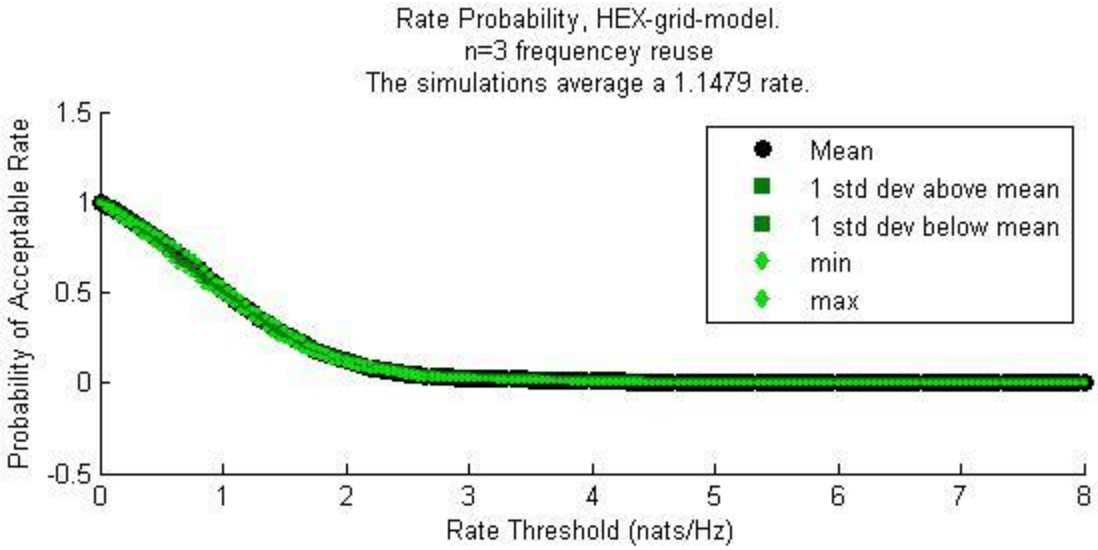


Figure 4.8 Data Rate of Traditional Reuse 3

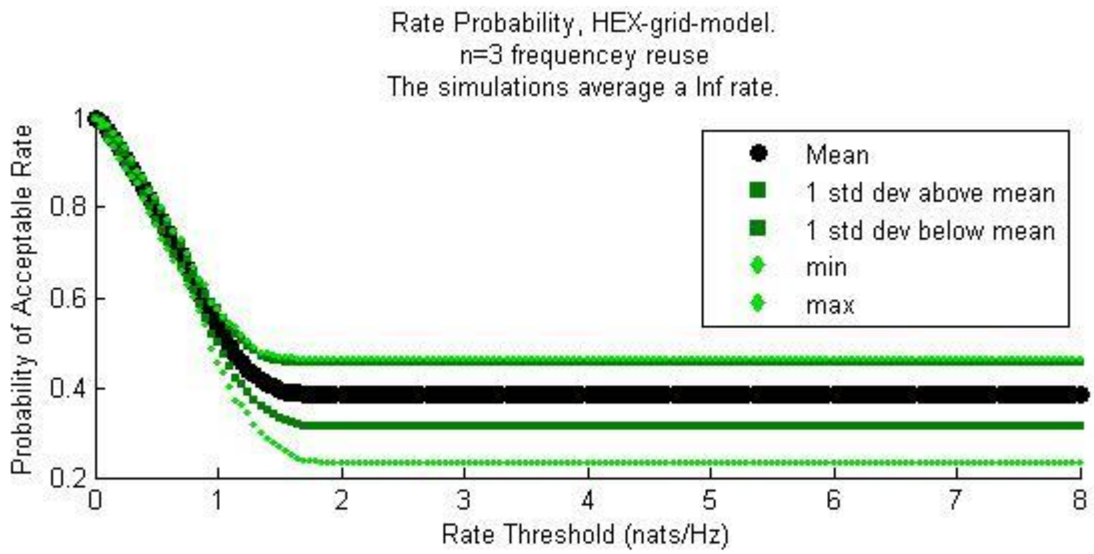


Figure 4.9 Data Rate of Fraction Frequency Ruse 3

4.2 Probability of Acceptance rate (PAR) calculation

S.N.	Rate (nats/Hz)	PAR1 Traditional Reuse1	PAR2 Traditional Reuse3	PAR3 FFR Reuse3	% gain w.r.t PAR1	% gain w.r.t PAR2
1	0	1	1	1	0	0
2	0.5	0.8455	0.8055	0.7533	-10.904	-6.480
3	1	0.6558	0.5836	0.4589	-30.024	-21.367
4	1.5	0.5176	0.2904	0.32928	-36.383	13.3
5	2	0.4065	0.2239	0.31015	-23.7	38.5
6	2.5	0.3279	0.1562	0.31015	-5.4	98.6
7	3	0.2547	0.1 274	0.31015	21.7	143.4
8	3.5	0.2141	0.01096 ~0	0.31015	44.862	31.02
9	4	0.1572	0.005479 ~0	0.31015	97.296	31.02
10	4.5	0.1192	0.00274 ~0	0.31015	160.1929	31.02
11	5	0.1003	0	0.31015	209.222	31.02
12	5.5	0.08401 ~0	0	0.31015	31.02	31.02
13	6	0.06775 ~0	0	0.31015	31.02	31.02

4.4 Data rate of to Traditional Reuses and FFR of Higher n=reuse size

Figure 4.11 and Figure 4.11 both were taken with n=9, at that 10 was built using integer reuse, and 11 was used with fractional reuse (with a Rate threshold of 0.8 nats/Hz and 10% of power & bandwidth allocated to the center).

In case of assumption situation in which some service requires about 1.3 nats/Hz to be covered. Perhaps it is some type of data functionality where constant use is not required. It is obvious that in this case, introducing fractional frequency reuse to the system has roughly 4 times of the amount of users whose can use this service at any one time.

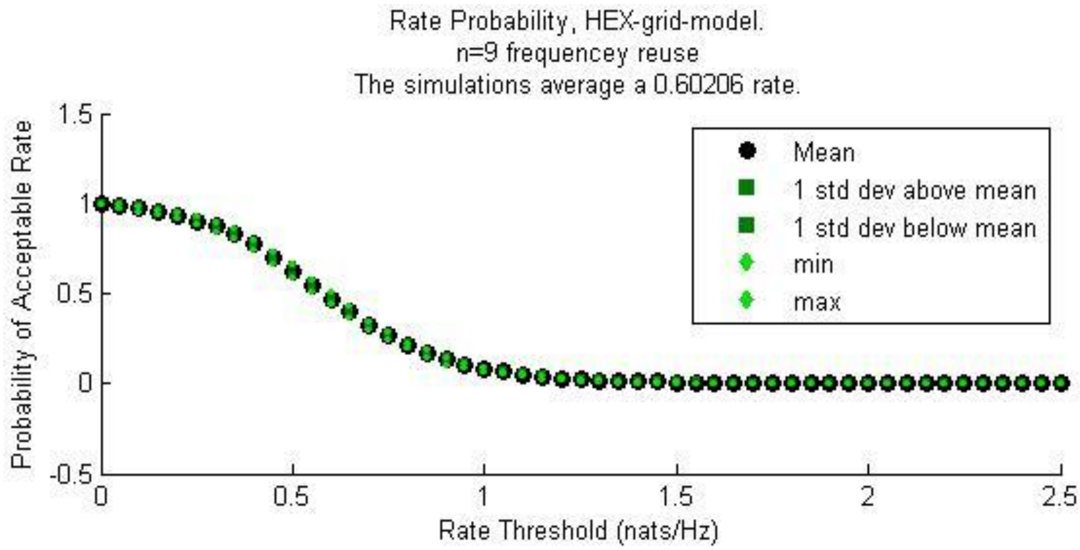
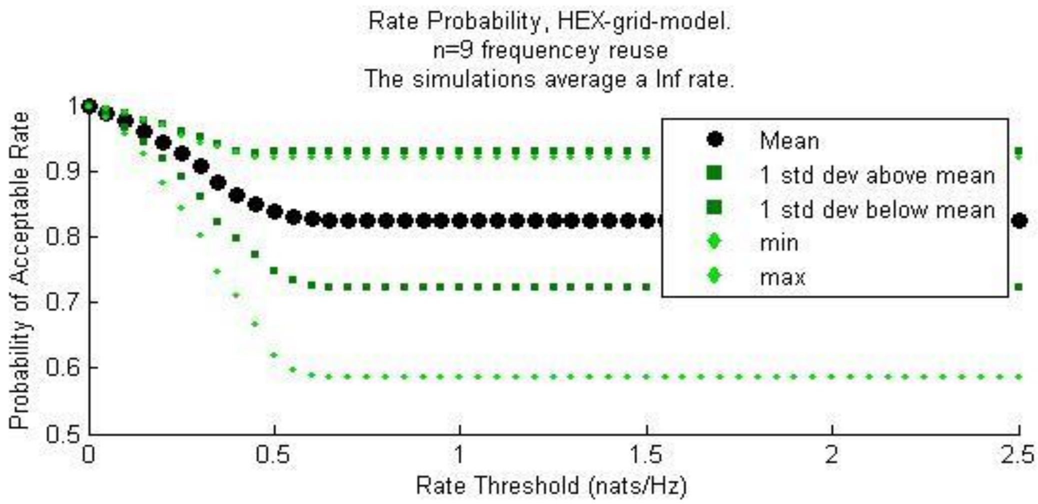


Figure 4.10 Data Rate of Traditional Reuse 9



4.11 Data Rate of Fraction Frequency Reuse 9

4.5 Comparing Reuse Factors for FFR

Figure4.13 shows that the increasing of the number of reuse assignments, SINR goes up (because it is harder for stations to interfere with one-another)

When low values of n = (1 – 4) are used the 80% achievable rate decreases monotonically while the 90%, 95% achievable rates see a slight rise before decreasing where the higher n gets not only the more SINR is available causing for a higher rate, but also at the same time, a higher n means less bandwidth for each sinfle mobile, thus a lower rate.

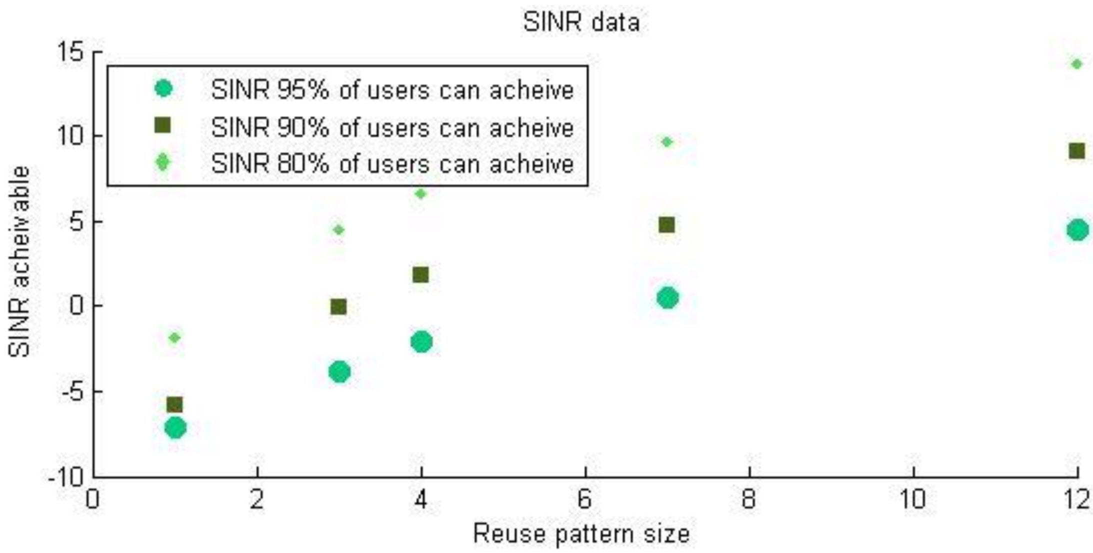


Figure 4.123 SINR of Fraction Frequency Reuse Lower Rf

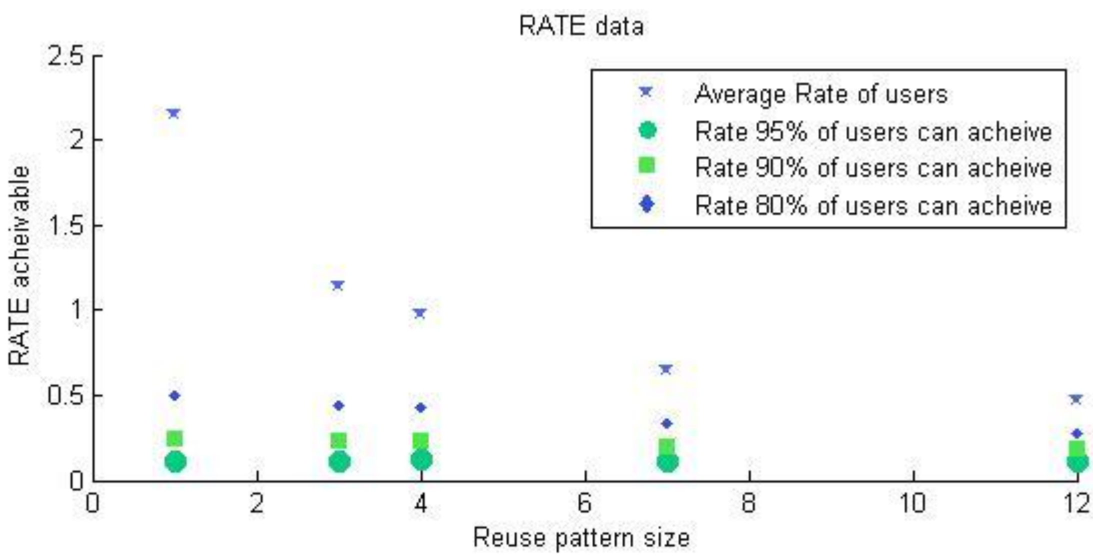


Figure 4.13 3 SINR of Fraction Frequency Reuse Higher Rf

Chapter Five

Conclusions and Recommendations

The main ideas presented in t3he thesis are collected and summarized in this chapter and recommendation for future work have been given in section two of this chapter.

5.1 Conclusions

This thesis evaluates interference management using FFR through Traditional Frequency reuse in 3GPP-LTE downlink homogenous condition show it is evaluated here. Results show that FFR provided better probability of coverage and probability of acceptance rate than traditional frequency reuse 1 and reuse 3. In fact, FFR balances the requirements of interference reduction and resources utilization efficiently, at that it has presented a new framework for downlink cellular network analysis, which significantly more tractable than the traditional grid-based models, and appears to track (and lower bound) a real deployment about as accurately as the traditional grid model (which upper bounds).

5.2 Recommendations

No single approach will in itself provide complete interference mitigation for an LTE implementation. Given the number of problems of contemporary interest that require modeling neighboring base stations, the possibilities for future work using this model are extensive. An extension to the uplink would be desirable. Further extensions to this approach could include random spatial placements of base stations that model repulsion.

A cohesive framework would allow for research into the dynamics and implications of FFR along with other important cellular network research including handoffs, base station cooperation, and FFR in conjunction with relays and/or femtocells.

5.3 References

[1] A. S. Hamza, S. S. Khalifa, H. S. Hamza, and K. Elsayed, "A survey on inter-cell interference coordination techniques in OFDMA-based cellular networks," *Communications Surveys & Tutorials, IEEE,* vol. 15, pp. 1642-1670, 2013.

[2] G. KAR, "PERFORMANCE OF STATIC AND ADAPTIVE SUBCHANNEL ALLOCATION SCHEMES FOR FRACTIONAL FREQUENCY REUSE IN WiMAX NETWORKS," BILKENT UNIVERSITY, 2011.

[3] L. F. a. X. Zhang, "Optimal fractional frequency reuse in OFDMA based wireless networks,", pp. 1–4. Wireless Communications, Networking and Mobile Computing, Crete Island, GreeceOctober 2008, pp. 1–4.

[4] I. A. Qaimkhani, "Improving Frequency Reuse and Cochannel Interference Coordination in 4G HetNets," 2013.

[5] R. K. G. Thomas David Novlan, Arunabha Ghosh, Jeffrey G. Andrews, "Analytical Evaluation of Fractional Frequency Reuse for OFDMA Cellular Networks," 26 Jan 2011 2011.

[6] F. R. P. Guillem Femenias Nadal "Performance of fractional frequency reuse (FFR) in 4G cellular networks."

[7] F. o. B. Jeffrey G. Andrews, and Radha Krishna Ganti, "A Tractable Approach to Coverage and Rate in

Cellular Networks," *arXiv:1009.0516v2 [cs.IT]*, 21 Feb 2011.

[8] I. A. Qaimkhani, "Dynamic Inter-Cell Interference Coordination in Cellular

OFDMA Networks," Master, Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada,, 2013.

[9] R. K. G. Harpreet S. Dhillon , Jeffrey G. Andrews, "A Tractable Framework for Coverage and Outage in Heterogeneous Cellular Networks," Feb. 07, 2010.

[10] H. S. Dhillon, R. K. Ganti, and J. G. Andrews, "A tractable framework for coverage and outage in heterogeneous cellular networks," in *Information Theory and Applications Workshop (ITA), 2011*, 2011, pp. 1-6.

[11] P. J. G. Andrews, "3G and 4G Cellular Standards " November 5 2008.

[12] J. Zolfaghari Bengar, "Design and performance evaluation of a radio resource management in OFDMA networks," 2013.

[13] E. J. Emil Björnson "Optimal Resource Management for Future Cellular and Heterogeneous Networks," 14 June 2014.

[14] F. Tariq, L. S. Dooley, A. S. Poulton, and Y. Ji, "Dynamic fractional frequency reuse based hybrid resource management for femtocell networks," in *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, 2011, pp. 272-277.

[15] L. Chen, "Coverage Planning and Resource Allocation in Broadband Cellular Access: Optimization Models and Algorithms," 2010.

[16] J. P. Gary Boudreau, Ning Guo, Rui Chang, Neng Wang, and Sophie Vrzic, Nortel. (April 2009) Interference Coordination and Cancellation for 4G Networks.

[17] J. G. Andrews, "Understanding Femtocell-Overlaid Cellular Networks," Dec. 6, 2010.

[18] K. Harrison and G. Ranade, "Dynamic Fractional Frequency Reuse in OFDMA systems," 2010.

[19] a. C. Chandra Thapa "Comparative evaluation of fraction frequency reuse (ffr) and traditional frequency reuse in 3gpp-lte downlink," *International Journal of Mobile Network Communications & Telematics,* vol. 1, 2012.

[20] M. Dzugan, "Simulating Effects of both Integer and Fractional Reuse on Achieved Rates in Modern Cellular Networks."

[21] E. Amaldi, A. Capone, and F. Malucelli. Planning UMTS base station location: optimisation models with power control and algorithms. *IEEE Transactions on Wireless Communications*, 2:939–952, 2003.

[22] A. R. Mishra. *Advanced Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G*. Wiley, 2007.

[23] G. Barriac and J. Holtzman. Introducing delay sensitivity into the proportional fair algorithm for CDMA downlink scheduling. In *Proc. of International Symposium on Spread Spectrum Techniques and Applications*, pages 652–656, 2002. [53] M. J. Nawrocki, M. Dohler, and A. H. Aghvami. *Understanding UMTS Radio Network: Modelling, Planning, and Automated Optimisation*. Wiley, 2006.

[24] LG Electronics. Interference mitigation in evolved UTRA/UTRAN. Technical report, 3GPP TSG RANWG1Meeting Documents, Reference number R1-050833, 2005.

[25] A. R. Mishra. *Advanced Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G*. Wiley, 2007.

Appendix A

```
%clear
clc
wb1 = waitbar(0,'Building Map...');

%% Assumptions
%
%   NumRings    -   Number of "rings" (including center) in grid
%   Lambda      -   Poisson intensity
%   alpha       -   Path loss exponent
%   R           -   Radius of one Hexagon Cell
%   frf_ij      -   frequency reuse vectors i and j
%   mu          -   Constant transmit power = 1/mu
%   SNRd        -   NOISE SNR = 1/(mu * sigma^2)         (in dB)
%   InnerRings  -   Number of "rings" on grid to consider (avoid edge-fx)
%   T_vals      -   Values of Threshold for SINR we will test
%   R_vals      -   Values of Threshol for Rate
%   SimDepth    -   Depth of simulation (number of runs)
%
                        NumRings    =   Params(1,1);
                        Lambda      =   Params(2,1);
                        alpha       =   Params(3,1);
                        R           =   1;
                        frf_ij      =   Params(4,1:2);
                        mu          =   Params(5,1);
                        SNRd        =   Params(6,1);
                        SNR         =   10^(SNRd/10);
                        InnerRings  =   Params(7,1);
                        T_vals      =   Params(8,1):Params(8,2):Params(8,3);
                        R_vals      =   Params(9,1):Params(9,2):Params(9,3);
                        SimDepth    =   Params(10,1);
                        draw_bounds =   Params(15,3);

                        PickFraction=   Params(11,1);




%% FRACTIONAL Reuse Parameters


%
%   P           -   number indicating percent designated to inner fraction
%                       ( P=0 simulates "Traditional" reuse)
%   P_sinr      -   SINR at which a mobile is moved to "center fraction
%                       ( P_sinr = inf simulates "Traditional" reuse)
%   P_rate      -   Rate at which a mobile is moved to "center fraction
%                       ( P_rate = inf simulates "Traditional" reuse)
%
%
%                           Available Spectrum
%
```

```
%        |___________________________________________________________|
%        |         0            |  1 |  2 |  3 |  4 |  5 |  6 |... | n  |
%        |_____________________|_____________________________________|
%        |                     P                                      |
%            designated for      split evenly by traditional reuse
%            inner fraction
%

                        P           =       Params(12,1);
                        P_sinr      =       Params(13,1);
                        P_rate      =       Params(14,1);

%
%
%            POWER_c                           POWER_r
%     power designated for center    power designated for reuse channels
%                    POWER_c + POWER_r = POWER_T (total P resources)
%
%    c_Pow       -    fraction of Total Power designated to Center Channel
%

                        c_Pow       =       Params(15,1);

                        POWER_T     =       1/mu;
                        POWER_c     =       (c_Pow)*POWER_T;
                        POWER_r     =       POWER_T - POWER_c;

P
P_sinr
P_rate
c_Pow
pause(3)

%% Initializations


    %    Build map assumption
                      .                             .

clc

frf_ij = sort(frf_ij,'descend');
frf_i = frf_ij(1,1);    frf_j = frf_ij(1,2);
frf_n = frf_i^2 + frf_j^2 + frf_i*frf_j;


Colors = zeros(frf_n,3);
for color = 1:frf_n;
    angle = color*2*pi / frf_n;
    Colors(color,:) = [.2+.1*rand .5+sin(angle)/2 .5+cos(angle)/2];
end
Colors = sortrows(Colors);
```

```
Base_Count = 0;
b_local_x = zeros(1,(2*(NumRings-1)+3)^2);
b_local_y = zeros(1,(2*(NumRings-1)+3)^2);
b_coord_i = zeros(1,(2*(NumRings-1)+3)^2);
b_coord_j = zeros(1,(2*(NumRings-1)+3)^2);
b_label   = zeros(1,(2*(NumRings-1)+3)^2);
for vert = -NumRings:NumRings;
    for diag = -NumRings:NumRings;
        Base_Count = Base_Count + 1;
        b_local_x(1,Base_Count) = (sqrt(3)*R)*cosd(30)*diag;
        b_local_y(1,Base_Count) = (sqrt(3)*R)*(vert + sind(30)*diag);
        b_coord_i(1,Base_Count) = vert;
        b_coord_j(1,Base_Count) = diag;

    end
end
CoordList = [b_coord_i' b_coord_j'];
```

```
F_m = zeros(6,2);
F_m(1,:) = [frf_i frf_j];
for F_r = 2:6;
    F_m(F_r,:) = [-F_m(F_r-1,2)   sum(F_m(F_r-1,:))];
end

F_ = F_m(1:2,1:2)';
for L = 1:frf_n;
   BC = 1;
   while (b_label(1,BC) ~= 0)&&(BC<Base_Count);
       BC = BC+1;
   end
   b_label(1,BC) = L;
   Zi = b_coord_i(1,BC);   Zj = b_coord_j(1,BC);    Z = [Zi Zj];
   for BC = 1:Base_Count;
       if b_label(1,BC)==0;
           diff = ([b_coord_i(1,BC)  b_coord_j(1,BC)] -  Z)';
           if norm(floor(F_\diff)-(F_\diff))<10^-8;
               b_label(1,BC)=L;
           end
       end
   end
end
```

```
b_local_d = sqrt(b_local_x.^2 + b_local_y.^2);
newCount = 0;
for BC = 1:Base_Count;
    if b_local_d(1,BC) <= sqrt(3)*R*(NumRings-1);
```

```
        newCount = newCount+1;
        B_x(1,newCount) = b_local_x(1,BC);
        B_y(1,newCount) = b_local_y(1,BC);
        B_l(1,newCount) = b_label(1,BC);
    end
end

Ax_11 = figure(1);
set(Ax_11,'visible','off')

waitbar(0,wb1,{'Building Map...';'This takes about 5-10 seconds'});

NumBase = length(B_x);
for BC = 1:NumBase;
    Bx = B_x(1,BC);     By = B_y(1,BC);
    if draw_bounds==1;
    line([Bx-R*cosd(60) Bx+R*cosd(60) Bx+R Bx+R*cosd(60) Bx-R*cosd(60)...
        Bx-R Bx-R*cosd(60)],[By+R*sind(60) By+R*sind(60) ...
        By By-R*sind(60) By-R*sind(60) By By+R*sind(60)]);
        hold on
    end
    fnb = floor(NumBase/4);
    if (BC == fnb)||(BC==2*fnb)||(BC==3*fnb);
    end
    scatter(Bx,By,70,Colors(B_l(1,BC),:),'filled');

end


NumBasei = 1;
for NBi = 1:InnerRings-1;
    NumBasei = NumBasei + 6*NBi;
end
B_dist = sqrt(B_x.^2 + B_y.^2);
Bdata = [B_dist' B_x' B_y' B_l'];  % distance % x % y % label
Bdata_s = sortrows(Bdata);
Bdata_in = Bdata_s(1:NumBasei , :);
scatter(Bdata_in(:,2)',Bdata_in(:,3)',20,'filled','w');




%% Simulations


%
%   P_cover     -   Probability of coverage
%   Rates       -   Average data rate per simulation run
%
                        P_cover     =   zeros(SimDepth,length(T_vals));
                        Rates       =   zeros(SimDepth,1);
```

```
%
%   We will also build the following arrays to store simulation information
%
%
%       Bdata_s                 Bdata_in                Mdata
%
%    dist x y Label           dist x y Label        dist x y Label d_B
%    dist x y Label           dist x y Label        dist x y Label d_B
%           .                        .                     .
%           .                        .                     .
%           .                        .                     .
%           .                 dist x y Label               .
%           .                                              .
%           .                                              .
%           .                                              .
%    dist x y Label                                        .
%                                                          .
%                                                          .
%                                                          .
%                                                   dist x y Label d_B
%
% Where     dist  is the object's distance from the origin  (col 1)
%           x     is the object's x coordinate              (col 2)
%           y     is the object's y coordinate              (col 3)
%           Label is a distinction on frequency for Reuse   (col 4)
%           d_B   is the mobile's distnace to its base      (col 5)
%

PsuedoR      =   700;
V_exprnd_g = exprnd(1,1,PsuedoR);
V_exprnd_h = exprnd(1/mu,1,PsuedoR);
V_exprnd_h_r = exprnd((POWER_r/POWER_T)*(1/mu),1,PsuedoR);
V_exprnd_h_c = exprnd((POWER_c/POWER_T)*(1/mu),1,PsuedoR);


waitbar(0,wb1,'Running Simulations');
tic
ClockStart = clock;
for sim = 1:SimDepth;




mobiles_x = zeros(1,Lambda*NumBasei);
mobiles_y = zeros(1,Lambda*NumBasei);

for MC = 1:Lambda*NumBasei;
    Theta = rand*2*pi;
    Radius = sqrt(rand)*(sqrt(3)*R*(InnerRings-1)+R);
    mobiles_x(1,MC) = Radius*cos(Theta);
    mobiles_y(1,MC) = Radius*sin(Theta);
end
```

```
NumMobiles = length(mobiles_x);
distances_to_bases = zeros(1,size(Bdata_s,1));
Mobile_Count = 0;
for n = 1:NumMobiles;
    Mx = mobiles_x(1,n);    My = mobiles_y(1,n);
    distances_to_bases = sqrt((Mx-Bdata_s(:,2)').^2+(My-Bdata_s(:,3)').^2);
    [val loc] = min(distances_to_bases);
    if (loc <= NumBasei);
        Mobile_Count = Mobile_Count+1;
        L = Bdata(loc,4);
        Mdata(Mobile_Count,:) = [sqrt(Mx^2+My^2) Mx My L val];
    end
end

if sim==1;
    scatter(Mdata(:,2)',Mdata(:,3)',3,'filled','k');
end
hold off




%% Coverage
%   We assume the SINR to be of the following form
%
%                 h * r^(-alpha)
%       SINR = -------------------
%               sigma^2 + I_r
%
%       h       -   exponentially distributed "rcv gain"
%       r       -   distance from user to his closest base station
%       alpha   -   path loss exponent
%       sigma^2 -   noise power
%       I_r     -   Cumulative Interference of all "other" base stations
%
%
%           where   I_r = SUM[ g * R^(-alpha)]
%                   over all base stations other than the closest
%
%                       g   -   Rayleigh fading factor
%                       R   -   Distance to base station
%

for q = 1:size(Mdata,1);
    h(q) = V_exprnd_h_r(ceil(PsuedoR*rand));
end
sigma_sq = 1 / (mu * SNR);
I_r = zeros(1,size(Mdata,1));
for m_ = 1:size(Mdata,1);   %for each mobile
    RunningSum = 0;
    for b_ = 1:size(Bdata_s,1); %for each base
        if Bdata(b_,4) == Mdata(m_,4); %when they are on same channel
            g = V_exprnd_g(ceil(PsuedoR*rand));
            R = ((Mdata(m_,2)-Bdata_s(b_,2)).^2 ...
            +(Mdata(m_,3)-Bdata_s(b_,3)).^2)^.5;

        if R == Mdata(m_,5);
```

```
            g = 0;
        end

        RunningSum = RunningSum + g*(R^(-alpha));
        end
    end
    I_r(1,m_) = RunningSum;
end


SINR_inner = (h.*((Mdata(:,5)').^(-alpha)))./(sigma_sq*(1-P)/frf_n + I_r);
SINR_dB = 10*log10 ( SINR_inner );



% Now we must go through each mobile again
% Every mobile with a high enough SINR/Rate can be put in the center band


if (PickFraction == 'S')||(PickFraction == 's');
    %%%%%%%%%%% Decide Based on SINR

    for m_ = 1:size(Mdata,1);   %for each mobile
        if(SINR_dB(1,m_) > P_sinr);
            if (sim==1);
                hold on
                scatter(Mdata(m_,2),Mdata(m_,3),'y')
                hold off
            end
            Mdata(m_,4) = 0; % 0 will be the label for CENTER
        end
    end

elseif (PickFraction == 'R')||(PickFraction == 'r');
    %%%%%%%%%%% Deicde Based on Rate

    for m_ = 1:size(Mdata,1);   %for each mobile
        if( log(1+SINR_inner(1,m_))*((1-P)/frf_n) > P_rate);
            if (sim==1);
                hold on
                scatter(Mdata(m_,2),Mdata(m_,3),'y')
                hold off
            end
            Mdata(m_,4) = 0; % 0 will be the label for CENTER
        end
    end

end
```

```
% we can now recaulculate the SINR
% knowing the new divisions in frequency allocation

% we start by reassinging the exponentially distributed power
% according to the previously defined divisions of POWER
for q = 1:size(Mdata,1);    %for each mobile
    if(Mdata(q,4) == 0 );
        h(q) = h(q)*(POWER_c/POWER_r);
    end
end


I_r = zeros(1,size(Mdata,1));
for m_ = 1:size(Mdata,1);    %for each mobile
    RunningSum = 0;
    for b_ = 1:size(Bdata_s,1); %for each base
        if Bdata(b_,4) == Mdata(m_,4); %when they are on same channel
            g = V_exprnd_g(ceil(PsuedoR*rand));
            R = ((Mdata(m_,2)-Bdata_s(b_,2)).^2 ...
            +(Mdata(m_,3)-Bdata_s(b_,3)).^2)^.5;

        if R == Mdata(m_,5);
            g = 0;
        end

        RunningSum = RunningSum + g*(R^(-alpha));
        end
    end
    I_r(1,m_) = RunningSum;


    if Mdata(m_,4)==0;
        Freq_Alloc_Percent = P;
    else
        Freq_Alloc_Percent = (1-P)/frf_n;
    end

    SINR_inner(1,m_) = ( h(1,m_).*( (Mdata(m_,5)).^(-alpha) ) )...
        ./(sigma_sq*Freq_Alloc_Percent + I_r(1,m_));
end


SINR_dB = 10*log10 ( SINR_inner );
QS(sim,:) = SINR_dB;




P_vals = zeros(size(T_vals));
for jj = 1:length(T_vals);
    P_vals(1,jj) = (sum(SINR_dB>T_vals(jj)))/(length(SINR_dB));
end
```

```
P_cover(sim,:) = P_vals;



%% Rate


%
%       Here we use Shannon's model for capacity to derive the rate
%
%           Tau (lambda,alpha)  =  E [ ln(1 + SINR) ]
%
%               Tau     -   average ergodic rate
%               lamda   -   Poisson intensity
%               alpha   -   Path loss exponent
%               E[ . ]  -   Expected Value operator
%               SINR    -   Signal to Interference-Noise Ratio
%
%                       SINR is calculated above for coverage
%


SINR_not_dB = 10.^((SINR_dB)./10);
for m_ = 1:size(Mdata,1);

    if Mdata(m_,4)==0;
        Freq_Alloc_Percent = P;
    else
        Freq_Alloc_Percent = (1-P)/frf_n;
    end

    Rate_per_user(1,m_) = (log(1+SINR_not_dB(1,m_)))*Freq_Alloc_Percent;

end
Rates(sim,1) = mean(Rate_per_user);





P_vals_rate = zeros(size(R_vals));
for jj = 1:length(R_vals);
    P_vals_rate(1,jj) = (sum(Rate_per_user>R_vals(jj)))/(length(SINR_dB));
end
P_cover_rate(sim,:) = P_vals_rate;

%% Post-Processing
%
%       display simulation progress
%       plot the distribution of coverage over varius Thresholds
%       Report the average rate of a typical user
%

percentg = sim/SimDepth;
ClockT = toc;
```

```
Remain = (ClockT/percentg) - ClockT;
Remain = round(100*Remain)/100;
waitbar(percentg,wb1,{'Running Simulations';...
    ['Im guessing another ',num2str(Remain), ' seconds']});
end
close(wb1)

%% Plots


Ax_22 = figure(2);
set(Ax_22,'visible','off')
P_cover_mean = mean(P_cover);
P_cover_std = std(P_cover);

scatter(T_vals, P_cover_mean,50,'filled','k');
hold on
scatter(T_vals, P_cover_mean-P_cover_std,20,'s','filled','b');
scatter(T_vals, P_cover_mean+P_cover_std,20,'s','filled','b');
scatter(T_vals, min(P_cover),20,'d','filled','c');
scatter(T_vals, max(P_cover),20,'d','filled','c');
title({'Coverage Probability, HEX-grid-model.';...
    ['n=', num2str(frf_n) , ' frequencey reuse' ]})
legend('Mean','1 std dev above mean','1 std dev below mean','min','max'...
    ,'Location','SouthWest');
xlabel('SINR Threshold (dB)');
ylabel('Probability of Coverage');
hold off




Ax_33 = figure(3);
set(Ax_33,'visible','off')

P_cover_rate_mean = mean(P_cover_rate);
P_cover_rate_std = std(P_cover_rate);

scatter(R_vals, P_cover_rate_mean,50,'filled','k');
hold on
scatter(R_vals, P_cover_rate_mean-P_cover_rate_std,20,[0 .5 0],'s','filled');
scatter(R_vals, P_cover_rate_mean+P_cover_rate_std,20,[0 .5 0],'s','filled');
scatter(R_vals, min(P_cover_rate),20,[0 .9 0],'d','filled');
scatter(R_vals, max(P_cover_rate),20,[0 .9 0],'d','filled');
title({'Rate Probability, HEX-grid-model.';...
    ['n=', num2str(frf_n) , ' frequencey reuse' ];...
    ['The simulations average a ' ,num2str(mean(Rates)), ' rate.']})
legend('Mean','1 std dev above mean','1 std dev below mean','min','max'...
    ,'Location','NorthEast');
xlabel('Rate Threshold (nats/Hz)');
ylabel('Probability of Acceptable Rate');
hold off
```

```
clc

Ax_1 = Ax_11;
Ax_2 = Ax_22;
Ax_3 = Ax_33;

QR=Rates;
Qs=P_cover_mean;
Qr=P_cover_rate_mean;

end
```

Appendix B

```
function varargout = Hex_gui_1_0(varargin)
% HEX_GUI_1_0 MATLAB code for Hex_gui_1_0.fig
%      HEX_GUI_1_0, by itself, creates a new HEX_GUI_1_0 or raises the existing
%      singleton*.
%
%      H = HEX_GUI_1_0 returns the handle to a new HEX_GUI_1_0 or the handle to
%      the existing singleton*.
%
%      HEX_GUI_1_0('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in HEX_GUI_1_0.M with the given input arguments.
%
%      HEX_GUI_1_0('Property','Value',...) creates a new HEX_GUI_1_0 or raises
the
%      existing singleton*.  Starting from the left, property value pairs are
%      applied to the GUI before Hex_gui_1_0_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property application
%      stop.  All inputs are passed to Hex_gui_1_0_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Hex_gui_1_0

% Last Modified by GUIDE v2.5 08-Jun-2011 08:54:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @Hex_gui_1_0_OpeningFcn, ...
                   'gui_OutputFcn',  @Hex_gui_1_0_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

% End initialization code - DO NOT EDIT
```

```
% --- Executes just before Hex_gui_1_0 is made visible.
function Hex_gui_1_0_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Hex_gui_1_0 (see VARARGIN)

% Choose default command line output for Hex_gui_1_0
handles.output = hObject;


set(handles.SimulationType,'SelectionChangeFcn',@SimulationType_SelectionChangeF
cn);
% Update handles structure
guidata(hObject, handles);

set(handles.ReuseType,'SelectionChangeFcn',@ReuseType_SelectionChangeFcn);
% Update handles structure
guidata(hObject, handles);

set(handles.FractionDecider,'SelectionChangeFcn',@FractionDecider_SelectionChang
eFcn);
% Update handles structure
guidata(hObject, handles);




% UIWAIT makes Hex_gui_1_0 wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = Hex_gui_1_0_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;




function NumberOfRings_Callback(hObject, eventdata, handles)
% hObject    handle to NumberOfRings (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NumberOfRings as text
%        str2double(get(hObject,'String')) returns contents of NumberOfRings as
a double
```

```
% --- Executes during object creation, after setting all properties.
function NumberOfRings_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NumberOfRings (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function InnerRings_Callback(hObject, eventdata, handles)
% hObject    handle to InnerRings (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of InnerRings as text
%        str2double(get(hObject,'String')) returns contents of InnerRings as a
double


% --- Executes during object creation, after setting all properties.
function InnerRings_CreateFcn(hObject, eventdata, handles)
% hObject    handle to InnerRings (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function Lambda_Callback(hObject, eventdata, handles)
% hObject    handle to Lambda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Lambda as text
%        str2double(get(hObject,'String')) returns contents of Lambda as a
double


% --- Executes during object creation, after setting all properties.
function Lambda_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Lambda (see GCBO)
```

```
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function Alpha_Callback(hObject, eventdata, handles)
% hObject    handle to Alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Alpha as text
%        str2double(get(hObject,'String')) returns contents of Alpha as a double


% --- Executes during object creation, after setting all properties.
function Alpha_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Alpha (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function SNR_Callback(hObject, eventdata, handles)
% hObject    handle to SNR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SNR as text
%        str2double(get(hObject,'String')) returns contents of SNR as a double


% --- Executes during object creation, after setting all properties.
function SNR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SNR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function NumRuns_Callback(hObject, eventdata, handles)
% hObject    handle to NumRuns (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of NumRuns as text
%        str2double(get(hObject,'String')) returns contents of NumRuns as a
double


% --- Executes during object creation, after setting all properties.
function NumRuns_CreateFcn(hObject, eventdata, handles)
% hObject    handle to NumRuns (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function SINRintL_Callback(hObject, eventdata, handles)
% hObject    handle to SINRintL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SINRintL as text
%        str2double(get(hObject,'String')) returns contents of SINRintL as a
double


% --- Executes during object creation, after setting all properties.
function SINRintL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SINRintL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function RATEintL_Callback(hObject, eventdata, handles)
% hObject    handle to RATEintL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RATEintL as text
%        str2double(get(hObject,'String')) returns contents of RATEintL as a
double


% --- Executes during object creation, after setting all properties.
function RATEintL_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RATEintL (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function SINRintINT_Callback(hObject, eventdata, handles)
% hObject    handle to SINRintINT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SINRintINT as text
%        str2double(get(hObject,'String')) returns contents of SINRintINT as a
double


% --- Executes during object creation, after setting all properties.
function SINRintINT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SINRintINT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function SINRintH_Callback(hObject, eventdata, handles)
% hObject    handle to SINRintH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SINRintH as text
%        str2double(get(hObject,'String')) returns contents of SINRintH as a
double


% --- Executes during object creation, after setting all properties.
function SINRintH_CreateFcn(hObject, eventdata, handles)
% hObject    handle to SINRintH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function RATEintINT_Callback(hObject, eventdata, handles)
% hObject    handle to RATEintINT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of RATEintINT as text
%        str2double(get(hObject,'String')) returns contents of RATEintINT as a
double


% --- Executes during object creation, after setting all properties.
function RATEintINT_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RATEintINT (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function RATEintH_Callback(hObject, eventdata, handles)
% hObject    handle to RATEintH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of RATEintH as text
%        str2double(get(hObject,'String')) returns contents of RATEintH as a
double


% --- Executes during object creation, after setting all properties.
function RATEintH_CreateFcn(hObject, eventdata, handles)
% hObject    handle to RATEintH (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on selection change in ReuseFactor.
function ReuseFactor_Callback(hObject, eventdata, handles)
% hObject    handle to ReuseFactor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns ReuseFactor contents
as cell array
%        contents{get(hObject,'Value')} returns selected item from ReuseFactor


% --- Executes during object creation, after setting all properties.
function ReuseFactor_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ReuseFactor (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



% --- Executes on slider movement.
function FractionSlider_Callback(hObject, eventdata, handles)
% hObject    handle to FractionSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
FractionDisplay = get(handles.FractionSlider,'Value');
set(handles.FractionSliderDisplay,'string',FractionDisplay);

% Hints: get(hObject,'Value') returns position of slider
```

```
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider


% --- Executes during object creation, after setting all properties.
function FractionSlider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FractionSlider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end



function ThresholdSINR_Callback(hObject, eventdata, handles)
% hObject    handle to ThresholdSINR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ThresholdSINR as text
%        str2double(get(hObject,'String')) returns contents of ThresholdSINR as
a double


% --- Executes during object creation, after setting all properties.
function ThresholdSINR_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ThresholdSINR (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


function SimulationType_SelectionChangeFcn(hObject, eventdata)
%retrieve GUI data, i.e. the handles structure
handles = guidata(hObject);
SimTypeTag =  get(eventdata.NewValue,'Tag') ;
set(handles.SimTypeID,'string',SimTypeTag);
if strcmp(SimTypeTag,'FixedReuse')
    set(handles.ReuseFactor,'Visible','on')
    set(handles.ReuseFactorText,'Visible','on')
elseif strcmp(SimTypeTag,'CompareReuse')
    set(handles.ReuseFactor,'Visible','off')
    set(handles.ReuseFactorText,'Visible','off')
end
```

```
guidata(hObject, handles);

function ReuseType_SelectionChangeFcn(hObject, eventdata)
%retrieve GUI data, i.e. the handles structure
handles = guidata(hObject);
ReuseTypeTag =  get(eventdata.NewValue,'Tag')  ;
set(handles.ReuseTypeID,'string',ReuseTypeTag);
if strcmp(ReuseTypeTag,'IntegerReuse')
    set(handles.FractionParametersPanel,'Visible','off')
    set(handles.FractionDeciderPanel,'Visible','off')
elseif strcmp(ReuseTypeTag,'FractionalReuse')
    set(handles.FractionParametersPanel,'Visible','on')
    set(handles.FractionDeciderPanel,'Visible','on')
end
guidata(hObject, handles);

function FractionDecider_SelectionChangeFcn(hObject, eventdata)
%retrieve GUI data, i.e. the handles structure
handles = guidata(hObject);
FractionDeciderTag =  get(eventdata.NewValue,'Tag');
set(handles.FractionDeciderID,'string',FractionDeciderTag);
if strcmp(FractionDeciderTag,'RATE_')||(strcmp(FractionDeciderTag,'Rate'));
    set(handles.ThresholdSINR,'Visible','off')
    set(handles.ThresholdSINRText,'Visible','off')
    set(handles.ThresholdRate,'Visible','on')
    set(handles.ThresholdRateText,'Visible','on')
elseif
(strcmp(FractionDeciderTag,'SINR_'))||(strcmp(FractionDeciderTag,'SINR'));
    set(handles.ThresholdSINR,'Visible','on')
    set(handles.ThresholdSINRText,'Visible','on')
    set(handles.ThresholdRate,'Visible','off')
    set(handles.ThresholdRateText,'Visible','off')
end
guidata(hObject, handles);




function ThresholdRate_Callback(hObject, eventdata, handles)
% hObject    handle to ThresholdRate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of ThresholdRate as text
%        str2double(get(hObject,'String')) returns contents of ThresholdRate as
a double


% --- Executes during object creation, after setting all properties.
function ThresholdRate_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ThresholdRate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on slider movement.
function FractionSlider2_Callback(hObject, eventdata, handles)
% hObject    handle to FractionSlider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider
FractionDisplay2 = get(handles.FractionSlider2,'Value');
set(handles.FractionSliderDisplay2,'string',FractionDisplay2);




% --- Executes during object creation, after setting all properties.
function FractionSlider2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FractionSlider2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end




% --- Executes on button press in Execute.
function Execute_Callback(hObject, eventdata, handles)
% hObject    handle to Execute (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



N       = str2num(get(handles.NumberOfRings,'String'));  %N
in      = str2num(get(handles.InnerRings,'String'));     %in
l       = str2num(get(handles.Lambda,'String'));         %l
alph    = str2num(get(handles.Alpha,'String'));          %alph
snr     = str2num(get(handles.SNR,'String'));            %snr
```

```
xx      = str2num(get(handles.NumRuns,'String'));          %xx
T_l     = str2num(get(handles.SINRintL,'String'));
T_i     = str2num(get(handles.SINRintINT,'String'));
T_h     = str2num(get(handles.SINRintH,'String'));
T_ = T_l:T_i:T_h;                                          %T_
R_l     = str2num(get(handles.RATEintL,'String'));
R_i     = str2num(get(handles.RATEintINT,'String'));
R_h     = str2num(get(handles.RATEintH,'String'));
R_ = R_l:R_i:R_h;                                          %P_

SimTypeTag = get(handles.SimTypeID,'String');             %SimTypeTag
ReuseTypeTag = get(handles.ReuseTypeID,'String');         %ReuseTypeTag

frf_nc  = get(handles.ReuseFactor,'String');
frf_n   = str2num(frf_nc{get(handles.ReuseFactor,'Value')});    %frf_n
P_sinr  = str2num(get(handles.ThresholdSINR,'String'));         %P_sinr
P_rate  = str2num(get(handles.ThresholdRate,'String'));         %P_rate
P_v     = str2num(get(handles.FractionSliderDisplay,'String')); %P_v
P_vP    = str2num(get(handles.FractionSliderDisplay2,'String'));%P_vP


count = 0;
for i = 1:10;
   for j = 0:i;
       count = count+1;
       reuse(count,:) = [i^2+j^2+i*j i j];
   end
end
reuse = sortrows(reuse);
frf_ij=0;
for i = 1:size(reuse,1)
    if reuse(i,1) == frf_n
        frf_ij = reuse(i,2:3);
    end
end


FrDecide = get(handles.FractionDeciderID,'String');
if (strcmp(FrDecide,'SINR'))||(strcmp(FrDecide,'SINR_'));
    PickFr = 'S';
else
    PickFr = 'R';
end




Params = zeros(15,3);
Params(15,3) = 1; %Draw Bounds ON

Params(1,1)   =     N;               %NumRings
Params(2,1)   =     l;               %Lambda
Params(3,1)   =     alph;            %alpha
Params(4,1:2) =     frf_ij;          %frf_ij
Params(5,1)   =     1;               %mu
Params(6,1)   =     snr;             %SNR (dB)
```

```
Params(7,1)    =     in;             %InnerRings
Params(8,1:3)  =     [T_l T_i T_h];  %T vals
Params(9,1:3)  =     [R_l R_i R_h];  %R vals
Params(10,1)   =     xx;             %SimDepth
Params(11,1)   =     PickFr;         %PickFraction
Params(12,1)   =     P_v;            %P
Params(13,1)   =     P_sinr;         %P_sinr
Params(14,1)   =     P_rate;         %P_rate
Params(15,1)   =     P_vP;           %c_pow


if strcmp(ReuseTypeTag,'IntegerReuse')
    %ok it's integer resue
    P_v = 0;
    P_vP = 0;
    P_sinr = inf;
    P_rate = inf;
    Params(12,1)   =     P_v;            %P
    Params(13,1)   =     P_sinr;         %P_sinr
    Params(14,1)   =     P_rate;         %P_rate
    Params(15,1)   =     P_vP;           %c_pow
elseif strcmp(ReuseTypeTag,'FractionalReuse')
    %ok it's fractional reuse
end


if strcmp(SimTypeTag,'FixedReuse');
    %ok it's fixed reuse
    %but which kind
    [im1 im2 im3 QS QR Qs Qr] = Hex_gui_function1(Params);
elseif strcmp(SimTypeTag,'CompareReuse')
    %ok we should compare reuse
    [im1 im2 im3] = Hex_gui_function2(Params);
end

scrnsz = get(0,'ScreenSize');
set(im1,'visible','on','OuterPosition',[1 1 scrnsz(4)/2.1 scrnsz(4)/2.1])
set(im2,'visible','on','OuterPosition',[1 floor(.55*scrnsz(4)) scrnsz(3)/2.1
scrnsz(4)/2.1])
set(im3,'visible','on','OuterPosition',[floor(.55*scrnsz(3))
floor(.55*scrnsz(4)) scrnsz(3)/2.1 scrnsz(4)/2.1])





% --- Executes on button press in Clear.
function Clear_Callback(hObject, eventdata, handles)
% hObject    handle to Clear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
for ijk=1:3
    %close all
    close(figure,ijk)
end
```

```
% --- Executes on button press in Keep.
function Keep_Callback(hObject, eventdata, handles)
% hObject    handle to Keep (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

f1 = figure(1);
f2 = figure(2);
f3 = figure(3);
KeepA = copyobj(f1,get(figure(3),'Parent'));
KeepB = copyobj(f2,get(figure(4),'Parent'));
KeepC = copyobj(f3,get(figure(5),'Parent'));


for ijk=1:3
    %close all
    close(figure,ijk)
end
```