# SELF-ORGANISATION IN LTE NETWORKS: AN INVESTIGATION

*by*

Andrew Thompson

Submitted for the degree of
Doctor of Philosophy

Department of Computer Science

School of Mathematical and Computer Sciences
Heriot-Watt University

February 2014

**Abstract**

Mobile telecommunications networks based on Long Term Evolution (LTE) technology promise faster throughput to their users. LTE networks are however susceptible to a phenomenon known as inter-cell interference which can greatly reduce the throughput of the network causing unacceptable degradation of performance for cell edge users.

A number of approaches to mitigating or minimising inter-cell interference have been presented in the literature such as randomisation, cancellation and coordination. The possibility of coordination between network nodes in an LTE network is made possible through the introduction of the X2 network link.

This thesis explores approaches to reducing the effect of inter-cell interference on the throughput of LTE networks by using the X2 link to coordinate the scheduling of radio resources. Three approaches to the reduction of inter-cell interference were developed. Localised organisation is a centralised scheme in which a scheduler is optimised by a Genetic Algorithm (GA) to reduce interference. Networked organisation makes use of the X2 communications link to enable the network nodes to exchange scheduling information in a way that lowers the level of interference across the whole network. Finally a more distributed and de-centralised approach is taken in which each of the network nodes optimises its own scheduling in coordination with its neighbours.

An LTE network simulator was built to allow for experimental comparison between these techniques and a number of existing approaches and to serve as a test bed for future algorithm development. These approaches were found to significantly improve the throughput of the cell edge users who were most affected by intereference. In particular the networked aspect of these approaches yielded the best initial results showing clear improvement over the existing state of the art. The distributed approach shows significant promise given further development.

# Acknowledgements

This research, funded by an EPSRC CASE studentship in collaboration with Motorola, was only possible with the help and support of others.

First and foremost I would like to thank my supervisor David Corne for his guidance throughout the process and for his initial suggestion that a PhD might be an interesting project; you were right! Thanks is also due to Chris Murphy of Motorola whose help in understanding both the wider context, and technical details of LTE were invaluable.

I would also like to thank my family and friends for their continued love and support. I'd particularly like to thank my parents who taught me to value knowledge, exploration and inquisitiveness, without which such research would be impossible. Finally, a special thanks to my wife Cara, for whose support, encouragement and patience throughout this process I am deeply grateful.

# Contents

# List of Tables

# List of Figures

# Acronyms

**3GPP** 3rd Generation Partnership Project.

**ARQ** Automatic Repeat reQuest.

**aSerFR** Adaptive Softer Frequency Reuse.

**aSerFR+X2** Adaptive Softer Frequency Reuse + X2.

**CDMA** Code Division Multiple Access.

**CQI** Channel Quality Indicator.

**dSerFR** De-centralised Softer Frequency Reuse.

**EA** Evolutionary Algorithm.

**eNB** Evolved NodeB.

**GA** Genetic Algorithm.

**ICIC** Inter-Cell Interference Coordination.

**IDMA** Interleave Division Multiple Access.

**LTE** Long Term Evolution.

**MCS** Modulation and Coding Scheme.

**MO-GA** Multi-Objective Genetic Algorithm.

**MOEA** Multi-Objective Evolutionary Algorithm.

**NB** NodeB.

**NGMN** Next Generation Mobile Networks Alliance.

**OFMD** Orthogonal Frequency Division Multiple.

**OFMDA** Orthogonal Frequency Division Multiple Access.

**PF** Proportional Fair.

**QoS** Quality of Service.

**RAN** Radio Access Network.

**RB** Resource Block.

**RE** Resource Element.

**RNC** Radio Network Controller.

**SerFR** Softer Frequency Reuse.

**SFR** Soft Frequency Reuse.

**SINR** Signal to Interfernce plus Noise Ratio.

**SON** Self-Organising/Optimizing Network.

**UE** User Equipment.

**UMTS** Universal Mobile Telecommunications System.

# Declaration Statement

There will be two blank pages here so I can add the pre-printed declaration form.

# Chapter 1

# Introduction

## 1.1 Introduction

This thesis describes an investigation into self-organisation in Long Term Evolution (LTE) networks. LTE is a wireless networking standard designed to enable high speed data connections for mobile phones and other devices. LTE networks typically consist of a number of mobile terminals (known as User Equipments (UEs)) each connected to a wireless access point (known as an Evolved NodeB (eNB)). These eNBs are connected both to other eNBs and crucially to the operator's core network which enables communication with other networks such as the internet.

The group responsible for standardising LTE networks, known as the 3rd Generation Partnership Project (3GPP), have identified self-organisation as a key way of enabling network operators to better manage the increasing complexity of their networks. It is envisaged that by incorporating self-organisation into the network nodes, such as the eNBs, a number of common network planning and optimisation tasks can be automated, allowing network operators to reduce their operating expenditure by reducing the manual involvement in such tasks [2]. For example, currently a large number of network parameters must be set and tuned over time by human intervention. It is envisioned that the use of self-organisation will enable the network itself to determine the optimal values for such parameters without the need for human intervention.

This chapter provides a high level introduction to LTE networks and self-organisation in order to provide the necessary information to understand the wider context of this thesis. The specific contributions made by this thesis are also highlighted.

## 1.2 Overview of LTE Networks

LTE networks are cellular wireless networks based on an existing technology known as Universal Mobile Telecommunications System (UMTS). UMTS is the current wireless network standard and has been widely adopted by the telecommunications industry. LTE is an evolutionary redesign of UMTS designed primarily to significantly increase the peak data rate (the maximum amount of information that can be transmitted), increase the performance of cell edge users (users that are located at the edge of a cell typically suffer from reduced data rates due to interference) and simplify the network architecture [1].

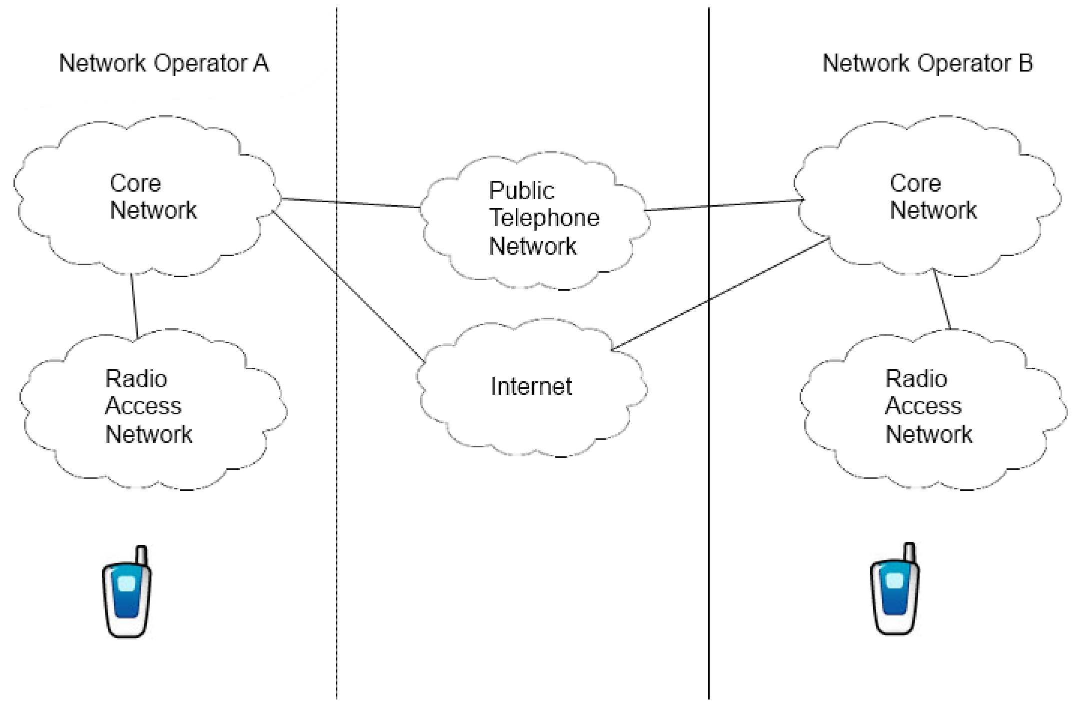


Figure 1.1: A conceptual representation of two wireless telecommunications networks, each of which is connected to both the public telephone network and the internet.

Conceptually wireless telecommunication networks can be viewed as consisting of two main elements: the radio access network and the core network, see Figure 1.1. The core network is the part of an operator's network infrastructure that provides systems for routing user calls and data sessions to their destinations either within the operator's network or to external networks. As such it maintains connections to both the internet, so that users can access standard web services, and to the public telephone network so that incoming and outgoing phone calls can be correctly routed. The core network is also responsible for many other functions such as authenticating users on their service, billing users for calls made and data sessions used, and providing services such as voice-mail.

The radio access network is that part of the network that wirelessly connects a user's device with the operator's core network. It is responsible for setting up and maintaining a wireless connection to any user devices and forwarding any calls or data sessions back to the core network for appropriate handling. The radio access network is typically implemented using a cellular network architecture as this allows a single radio access network to cover a significant area.

One of the main reasons self-organisation is being considered for use in LTE networks is due to changes in the design of the radio access network. To understand the significance of these changes it is first important to understand how radio access networks and more specifically cellular networks operate.

This section presents an overview of the cellular network architecture adopted by the 3GPP for use in the radio access network within UMTS and LTE and highlights some of the main differences in UMTS and LTE which have led to the possibility of implementing Self-Organising/Optimizing Network (SON) techniques.

### 1.2.1 Radio Access Network Architecture

Radio access networks are used to provide a wireless link between users' devices and the network operator's core network and typically take the form of a cellular network.

Cellular networks aim to provide connectivity for users (known as UEs) over a geographic area by using radio waves to transmit and receive information. The total area to be covered by the network is divided up into much smaller areas known as cells, each of which is served by a single base station.

Each base station is equipped with a number of antennae with which it communicates to the UEs within its cell and is connected to the operator's core network allowing it to route calls and data sessions made by the UEs in its area to their appropriate destination in other parts of the network. These cells can have a range of sizes depending on their locality and setup. For example a cell radius of 5km would be common in rural areas where the population is less dense, while in an urban environment cells would typically have a radius of 1km or less. Typically these cells are represented as hexagons as in Figure 1.2.

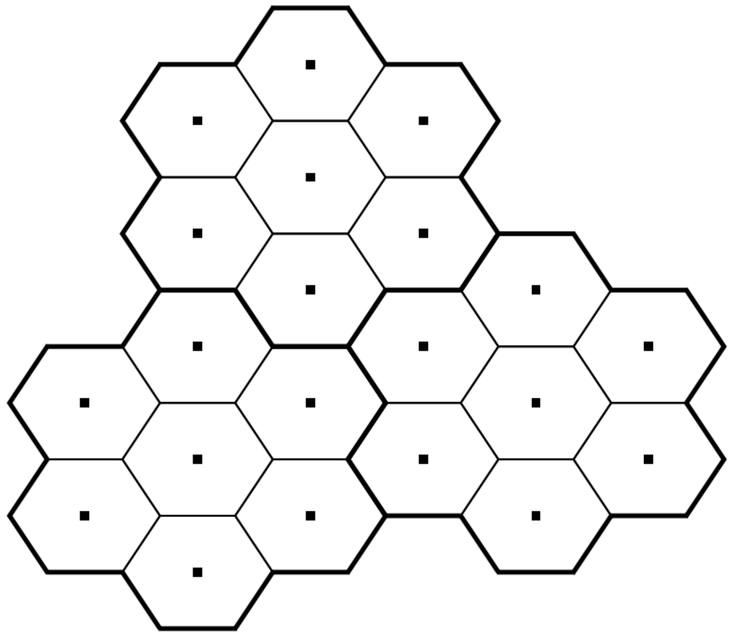

Figure 1.2: A simple cellular network arranged in 3 clusters, each with 7 cells.

The cellular architecture allows for a greater coverage area to be served by the network. The fact that any wireless transmission inevitably loses strength as it travels through space means that any base station will have an inherently limited range. Thus by dividing the desired coverage area into a number of smaller cells, each of which has a limited range, a much larger area can be covered [84, 94]. The cellular system has the additional benefit of increasing the ease with which the network can be expanded by the operator. To increase the size of the coverage area additional cells can be created by adding more base stations.

Another compelling reason for the use of a cellular system architecture is that it allows for the reuse of the available spectrum. Network operators are typically granted a license to transmit on a specific and limited range of frequencies. The frequency spectrum is a finite resource and thus there are a limited number of radio channels available to wireless systems [56]. This means that the total bandwidth of the system, and thus the amount of data that it can transmit, is dictated by the amount of available spectrum. Since the capacity of the network is constrained by the available spectrum, re-using that spectrum can greatly increase the capacity of the network.

Traditionally a cellular system allows the available spectrum to be divided into a number of bands, each of which can then be allocated to a cell. These frequency bands can then be reused in other cells provided that there is sufficient distance to prevent any interference. By dividing a large network into a number of smaller clusters the spectrum can be reused across the network in a predictable way. For example Figure 1.3 shows a small network composed of 3 clusters, each with 7 cells. The available spectrum has been divided into 7 bands, labelled $F1$ to $F7$, which have then been allocated to each cluster in a similar fashion.

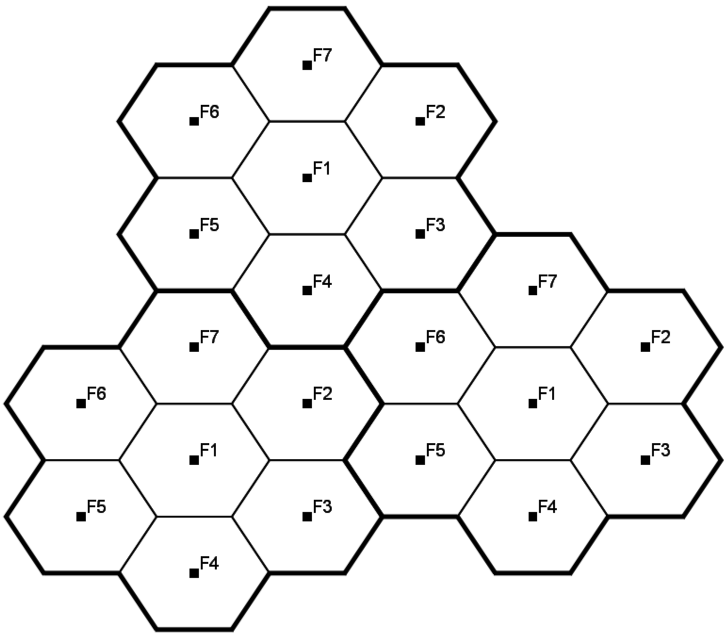


Figure 1.3: A simple cellular network demonstrating frequency reuse

Each cell can be further subdivided into sectors, typically three or four, each of which is served by a distinct directional antenna on the base station. Dividing an ideal cell into $N$ sectors should increase the capacity by $N$, however there are costs involved such as the increased cost of equipment and signalling overhead due to the need for handovers. A handover procedure occurs when a UE moves from one cell or sector to another. In order to correctly route incoming calls and data packets the network must be able to detect when a UE has moved from one cell or sector to another. Handovers typically reduce the throughput experienced by a UE and put additional pressure on the base station's resources. Sectorising cells can also reduce both the co-channel interference within the cell and the interference caused by the transmissions of neighbouring cells [25] and can lead to increased frequency reuse as shown in Figure 1.4, allowing several frequencies to be used within a single cell. Within UMTS networks this has been shown to significantly increase the capacity of cells [62].

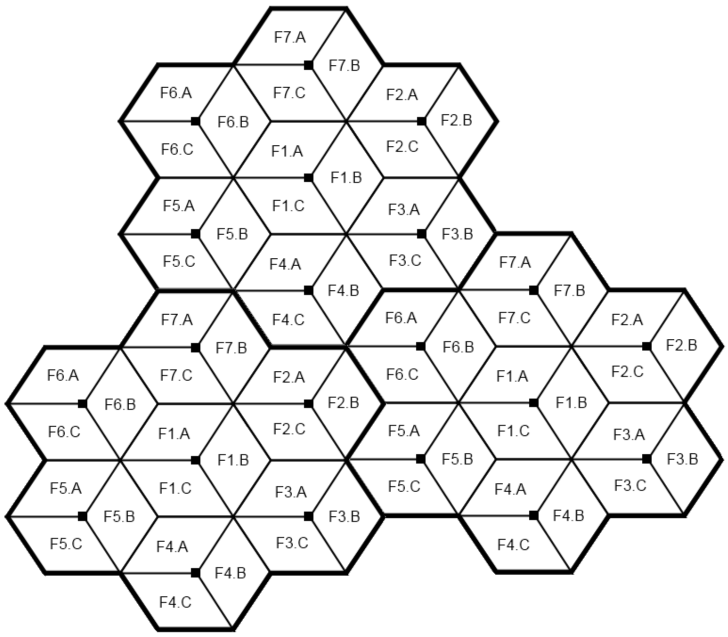


Figure 1.4: A simple cellular network demonstrating both frequency reuse and sectorisation.

### 1.2.2 UMTS Radio Access Network Architecture

The radio access network utilised by UMTS takes the form of a cellular network and introduces additional network infrastructure as shown in Figure 1.5. As with traditional cellular networks each cell is served by a base station, known as a NodeB (NB) in UMTS terminology, which is responsible for sending and receiving the actual wireless transmissions to and from the UEs in its cell. However a new network node, known as the Radio Network Controller (RNC), is introduced.

Each NB in a UMTS network is controlled by a single RNC, but a single RNC will control many NBs. The exact number of NBs controlled an individual RNC

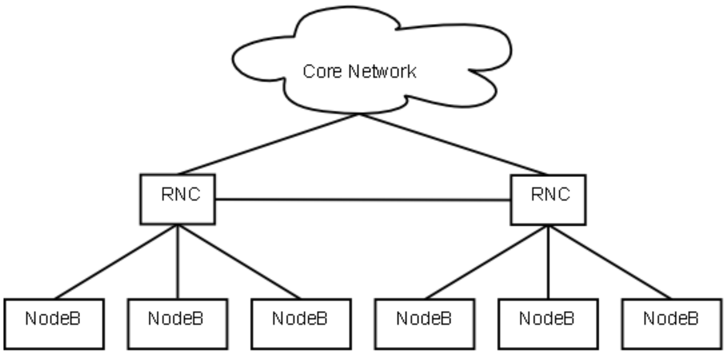


Figure 1.5: UMTS radio access network architecture

varies across deployments, but several hundred is not uncommon.

The RNC is responsible for a number of important functions including setting up the various connections and resources required to initiate an incoming or outgoing call or data session, the handling of Quality of Service (QoS) requirements for on-going connections and the management and allocation of the radio resources within each of the cells that it controls. The RNC is also responsible for implementing the Automatic Repeat reQuest (ARQ) protocol, which handles the re-transmission of erroneous or missing data, on all connections.

All handovers are handled by the RNC since NBs cannot communicate directly to other NBs. If a UE moves from a cell controlled by one RNC to a cell controlled by another RNC the RNCs will establish a connection in order to coordinate the transfer of any ongoing connections.

Most of the processing power within the UMTS radio access network is located in the RNC to the extent that the NB is little more than a modem that connects the UEs to the RNC.

### 1.2.3 LTE Radio Access Network Architecture

The radio access network in LTE is an enhanced version of that found in UMTS and as such shares many similarities.

The most significant difference, in the context of this research, is the removal of the RNC node, see Figure 1.6. The functionality and roles which were previously fulfilled by the RNC have now been devolved into the NBs, which have been renamed to Evolved NodeBs (eNBs) to reflect their more autonomous status. The eNBs are now directly responsible for assigning and scheduling radio resources, dealing with handovers to other eNBs and routing calls and data connections to their destination.

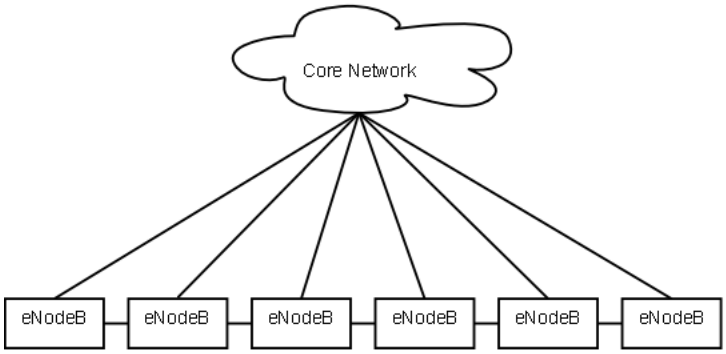


Figure 1.6: LTE radio access network architecture

Another important change in LTE is the addition of the X2 interface. Each eNB is directly connected to the core network and can also set up and maintain direct connections to other eNBs in the network using the X2 protocol [18]. This allows the eNBs to coordinate directly with each other regarding decisions such as handovers and scheduling rather than requiring the involvement of the RNC. It is this change in architecture which has led to an increasing interest in self-organisation since each eNB can be thought of as an independent agent in a distributed system.

Another change in the radio access network in LTE is the adoption of Orthogonal Frequency Division Multiple Access (OFMDA) as the transmission scheme. The transmission scheme determines how the data to be transmitted is actually encoded and transmitted using the radio wave as the transmission medium. The particular benefit of OFMDA, in the context of this thesis, is the fact that it essentially eliminates intra-cell interference. This means that there is very little interference within a single cell from the transmissions sent from that eNB. The immediate benefit of this is that the whole available spectrum can be re-used in each cell or even each sector within each cell. Since the amount of bandwidth within a system is directly proportional to the available spectrum this means that LTE networks have significantly more bandwidth available to them, thus potentially being able to provide high data rates and throughput.

### 1.2.4 Transmission Scheduling in LTE

Much of the detail of this research is concerned with the scheduling of transmissions within LTE networks and as such this section provides a general overview of how transmissions take place in LTE networks. The details of wireless transmission in LTE networks is contained in a number of standards documents authored by 3GPP, primarily [15], but with other details contained in [16, 17, 10, 3]. Their relevant content is summarised here since it provides important context to this research.

The resources that must be scheduled in an LTE system are the physical resources that are used to transmit the data to and from the UEs. These resources have both a time dimension and a frequency dimension.

The time domain is sub-divided in a number of different ways as seen in Figure 1.7. The largest division is known as a radio frame and has a duration of 10ms. Within each radio frame are ten subframes each with a duration of 1ms and each being divided into two further divisions known as slots with a duration of 0.5ms each. Finally, each slot can also be divided into 7 Orthogonal Frequency Division Multiple (OFMD) symbols. These symbols are essentially the physical wave-forms that are used to encode and transmit the actual data to be sent across the wireless channel.

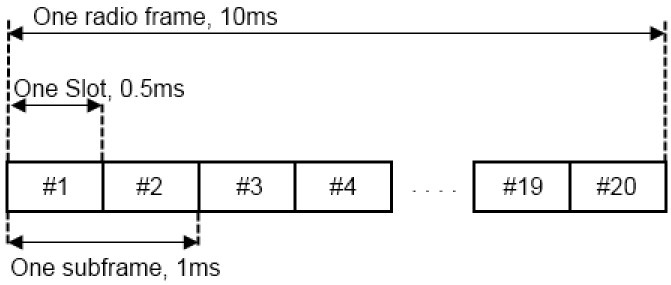


Figure 1.7: The time structure of the LTE physical resource.

In the frequency domain the physical resource is divided into a large number of sub-carriers. The exact number of sub-carriers in an LTE system depends mainly on transmission bandwidth (see Table 1.1), but can vary between 72 and 1200.

The basic scheduling unit in LTE is known as a Resource Block (RB). In the time domain a RB spans 0.5ms (or one slot), meaning that there are two RBs per sub-frame in the time domain. In the frequency domain a RB spans 12 sub-carriers. The actual number of RBs varies based on the network operator's configuration but can vary between 6 and 100, see Table 1.1. Figure 1.8 shows the typical breakdown of the frequency domain in LTE.

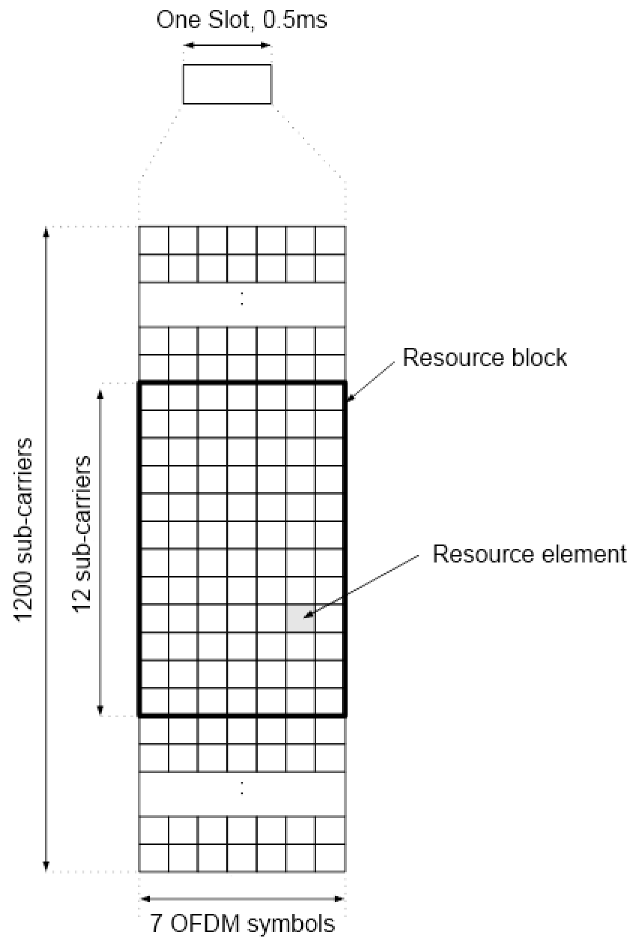


Figure 1.8: The frequency structure of the LTE physical resource.

The amount of data that can be transmitted in an RB is dependent on the modulation and coding scheme used. Modulation and coding schemes are essentially a set of parameters, such as modulation type and code rate, that determine how the data to be transmitted is encoded on the physical medium. Different modulation and coding schemes are better suited to different radio conditions. For example when the radio conditions are good, that is there is little interference, a more efficient modulation and coding scheme can be used, one that maximises the amount of data transmitted at the expense of error checking codes and redundant data. However in more adverse radio conditions a modulation and coding scheme that sacrifices the total amount of data for additional error checking codes and redundancy will perform better. The most optimistic modulation and coding scheme can transmit 6 bits per OFMD symbol.

| Channel Bandwidth (MHz) | 1.4 | 3 | 5 | 10 | 15 | 20 |
|---|---|---|---|---|---|---|
| Maximum number of sub-carriers | 72 | 180 | 300 | 600 | 900 | 1200 |
| Maximum number of RBs | 6 | 15 | 25 | 50 | 75 | 100 |

Table 1.1: The relationship between channel bandwidth, number of sub-carriers and the number of available RBs

During each slot an eNB may schedule an RB to any of the UEs within its cell. A scheduler typically tries to allocate each RB to a UE such that the maximum amount of data can be transmitted to that UE. There are however a number of reasons why this is not always possible. The quality of the wireless channel between the eNB and UE is constantly changing due to interference, movement and background noise. The quality of the wireless channel dictates how much data can be transmitted to the UE based on the modulation and coding scheme.

The scheduling of RBs can be thought of as an optimisation problem in the sense that there are typically two competing requirements: maximising the throughput of the eNB while also ensuring that all UEs are able to maintain adequate data rates. For example if an eNB was simply to maximise its throughput it could schedule as many RBs as possible to those UEs with the best wireless channel conditions, this would maximise the amount of data transmitted in each RB since an optimistic modulation and coding scheme could be used. But this would result in those UEs experiencing poor wireless channel conditions to be starved of data. Thus a good scheduler must ensure that those UEs with poor channel conditions receive adequate data.

Another important factor in scheduling is the fact that transmissions from one eNB can cause significant interference with the transmissions from neighbouring eNBs. This is known as inter-cell interference. Thus the scheduling decisions made by one eNB can have a dramatic effect on the performance of UEs being served by other eNBs. In order to prevent or, more realistically, minimise this interference a number of Inter-Cell Interference Coordination (ICIC) schemes have been developed. These schemes generally use a combination of frequency allocation schemes and schedulers to reduce the level of inter-cell interference and it is expected that within this context self-organisation can offer significant improvements.

## 1.3 Self-Organisation

Within the context of LTE networks, self-organisation is concerned with automating some aspect of the network's setup, optimisation or maintenance which was, in previous networks, a manual process. Both 3GPP and Next Generation Mobile Networks Alliance (NGMN) have published a list of use cases [7, 23] which highlight a number of areas where self-organisation is expected to provide significant benefits. Generally this takes the form of the automatic adaptation of network parameters so that some aspect of the performance of the network is improved.

More generally, self-organisation can be thought of as "the mechanism or the process enabling a system to change its organisation without explicit external command during its execution time" [42]. The essential concept for this comes from the natural realm where numerous examples can be found, particularly of systems where beneficial global behaviour emerges from simple interactions. For example groups of birds are often seen displaying a level of self-organisation in their flocking behaviour. This co-ordination of the movement of a large number of birds is achieved without any central control, but merely by each bird following a number of simple rules.

Within computer science the ideas behind these natural systems have been used as the basis of a number of problem solving techniques. For example capturing the rules governing the flocking behaviour of birds [83] has been used to accurately simulate the motion of flocking animals for animation. The foraging behaviour of ants has also been adapted as an effective way of solving traditional graph based problems [45] for example the travelling sales man problem [44]. Other examples of self-organisation found in the biological world which have found applications in computer science include brood sorting, web weaving and nest building [71] which have been applied to areas as diverse as database organisation and clustering, network routing and task cooperation in robots.

The focus of this research is how network performance in LTE can be improved by utilising the interactions between neighbouring cells. The addition of the X2 interface to LTE networks, which allows for direct communication and interaction between eNBs, may enable the eNBs to respond and adapt to their network environment in a more organic and de-centralised way.

## 1.4 Contributions

This thesis offers a number of contributions in the area of self-organisation with application to LTE networks in particular.

- The first contribution of this thesis is the introduction of localised organisation into the scheduling of transmissions in LTE networks, that is organisation that is local to each eNB with no interactions with other eNBs. A parameterised scheduling algorithm was developed that can be optimised to different network environments using a Genetic Algorithm (GA). Based on simulations on a number of network environments using standard parameters this approach was seen to significantly improve on existing scheduling algorithms particularly on networks which experienced a mixed traffic load.

- The second contribution made by this thesis is the introduction of networked organisation into the scheduling of transmissions in LTE networks, that is organisation that involves co-ordination between eNBs. A network aware scheduler was developed that enabled communication and collaborative scheduling among eNBs using the X2 link. This collaborative algorithm was designed such that it can be used with any existing scheduling algorithm to provide improved performance. The algorithm was parameterised in a way that enables a GA to optimise its performance for different networked environments. As with the localised organisation a number of simulations were run to compare this approach to existing techniques. These showed that under a number of network conditions particularly when there were fewer UEs or when traffic was light, this algorithm was able to provide significantly improved performance.

- The third contribution made by this thesis is in the use of distributed organisation to further improve the scheduling of transmissions in LTE networks. This builds on the previous contributions by enabling the performance optimisation to be done in a distributed, per-eNB basis rather than on a network-wide level. This allows each eNB to optimise the scheduling for its own specific environment, while also collaborating with other eNBs. In order to compare this approach with both the previous approaches and other existing approaches simulations were again run which showed that this algorithm provided performance gains across a number of network environments.

- Another contribution made by this thesis is the creation and release of a physical layer LTE simulator. This simulator is capable of simulating the downlink channel of an arbitrary number of eNBs and UEs with a number of different traffic levels, user speeds and scheduling algorithms. The release of this simulator should speed further research efforts in this area by avoiding the need to re-implement such a simulator. The source code for the simulator is available at `http://downthewire.co.uk/projects/jLTE-Simulator/`.

- The final contribution of this thesis is the direct comparison of a number of different scheduling algorithms for LTE. Five existing schedulers were compared to the novel schedulers presented as part of this work. Each of the schedulers was compared across twenty-seven different network environments which included variations in the number of UEs, the traffic level of the UEs and the speed of the UEs. The performance of each of the schedulers is compared across two main axes allowing for the performance of each scheduler to be compared across network conditions.

## 1.5 Outline of Thesis

The remainder of this thesis is divided into the following sections.

Chapter 2 provides a more detailed overview of LTE networks, in particular the problem of inter-cell interference which is the main focus of this research. A number of existing solutions are described and contrasted. Chapter 3 details a self-organising method developed which operates locally to each eNB without coordination from other eNBs. The working of this algorithm along with some comparative experiments were used to show that this kind of optimisation can provide some level of performance improvement over the existing approaches. A network simulator used as the basis for the experiments is introduced and described. Chapter 4 reports on a more coordinated approach to self-organisation in which eNBs communicate with each other using the X2 link to achieve significantly improved performance. Chapter 5 explores a more distributed approach to the ICIC problem by allowing eNBs to continuously evolve their own configuration based on their local network environment. Chapter 6 reviews the work that has been done and also offers some suggestions for areas where further study would provide additional benefits.

# Chapter 2

# Literature Review

## 2.1 Introduction

Long Term Evolution (LTE) is an emerging wireless networking standard. As with any new wireless technology, networks built using LTE face numerous optimisation challenges. A particularly important problem in LTE networks is that of interference. Since this research is focused on developing coordinated approaches to mitigate the effects of interference, a number of optimisation problems which contribute to the prevalence of interference in a network are first discussed.

The research described in this thesis presents a number of approaches to tackling the interference problem in LTE networks using principles of self-organisation and coordination. In particular Evolutionary Algorithms (EAs) in the specific form of Genetic Algorithms (GAs) provide the means of self-organisation and are introduced in Section 2.5.

## 2.2 Optimisation issues in LTE affecting Interference

The design and optimisation of a wireless network is an on-going process throughout the life-cycle of the network. It is concerned with finding the network setup which maximises performance while usually minimising cost to the operator [75]. Some aspects of the network's setup must be decided upon before the network is deployed and must remain fixed for the lifetime of the network. For example, the placement of Evolved NodeBs (eNBs) is generally determined prior to network deployment as it is expensive and impractical to move an existing eNB. Other aspects of the network setup may be adjusted after the network had been deployed. For example cell size, Quality of Service (QoS) constraints and some of the parameters of the antennae of an eNB can all be adjusted to optimise the network after deployment. This section presents some common optimisation problems faced by wireless networks such as LTE.

### 2.2.1 Base Station Placement

As discussed, an LTE network consists of a number of fixed eNBs which serve the users within their radius. In deploying an LTE network, operators generally want to provide connectivity for the largest possible continuous area for the least possible cost. This constraint requires that the number of eNBs used be kept to a minimum since they represent a significant financial cost.

LTE networks are designed to provide mobile access which means that users or User Equipments (UEs) must be able to move from cell to cell while still maintaining any existing calls or data sessions. It is therefore important that the coverage area provided by the eNBs is contiguous.

A simple LTE network is depicted in Figure 2.1. In moving directly from cell 1 to cell 3 UE1's network connection will be dropped due to a lack of overlapping coverage between the cells. This will result in an interruption of the UE's service and the dropping of any data connections or voice calls. In order to prevent this there must be some area of overlap between the cells so that the UE can pass from one cell to another without dropping the connection. For example UE2 can move freely between cell 2 and cell 3 without dropping its connection as there is sufficient coverage between the cells. However this overlapping area of coverage also introduces interference between the cells which degrades their performance and so should be kept to a minimum. This is discussed more fully in Section 2.3.2.

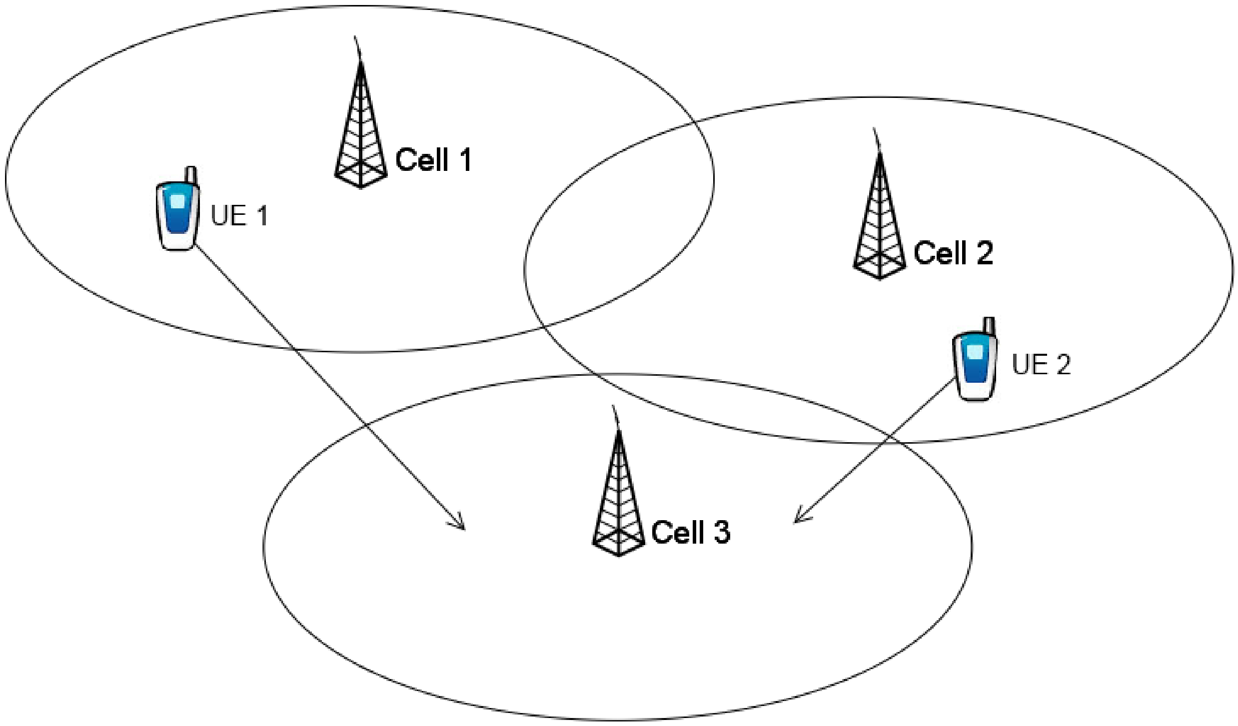


Figure 2.1: The importance of base station positioning for the movement of UEs between cells.

Another factor which should be taken into account is the expected density of UEs, and thus the expected traffic level, in the area of the cell. Each cell has a finite amount of resources at its disposal and so it can only provide a certain level of performance to the UEs that it is serving. If the number of UEs per eNB is too high then the UEs will experience poor levels of connectivity. Due to this an urban centre will have a much greater density of UEs, resulting in smaller cell sizes and thus more eNBs, whereas cells can be much larger in rural areas and thus fewer eNBs are required.

Several GAs have been developed in order to optimise the placement of eNBs. The multi-objective GA proposed in [97] explicitly balances the network coverage provided by the eNBs against the cost of the equipment, while [81] describes a GA which finds near optimal placement for the eNBs of an unnamed 3rd generation wireless network taking into account both traffic density and terrain. A flexible encoding scheme is used which allows the genetic operators to alter both the number and locations of the eNBs. However this is not practical as eNBs can often only be placed at specific places such as the roof of a building. The GA was shown to provide over 90% cover in both homogeneous and, more realistically, in heterogeneous traffic environments. As an alternative to GAs the Adaptive search algorithm developed in [24] provides good solutions to this problem for medium and large Universal Mobile Telecommunications System (UMTS) networks.

### 2.2.2 Antenna Set-up

The physical parameters of the antennae used by the eNBs have a significant effect not only on the performance of the cell but also on the level of interference experienced by the network. These parameters can be optimised early in the network's life-cycle and are often combined with the eNB placement problem. For example the multi-objective EA developed in [73] attempts to find both the optimal eNB locations along with the optimal antenna settings. However given a set of fixed eNBs, various aspects of the antenna set-up can still be optimised.

There are a number of primary antenna parameters which can significantly impact the performance of the network.

**Antenna Downtilt** The antenna downtilt is the angle of the main beam of an antenna below the horizontal plane. This angle is determined by a combination of the mechanical downtilt and electrical downtilt. These attributes should be considered separately, as opposed to modelling the total combined downtilt, due to the fact that they affect the coverage of the antenna in different ways [53, 66, 80]. Mechanical downtilt involves altering the angle of the antenna on adjustable brackets whereas electrical downtilt allows the antenna tilt to be changed without the need to physically adjust the angle. This is done by altering the strength of the currents in the elements of the antenna which causes the radiation pattern of the antenna to be altered [91].

**Azimuth** The azimuth of the antenna is the horizontal angle between north and the direction of the antenna. It can have a significant impact on both the coverage provided by the antenna and on the sector overlap. Typically altering the azimuth must be done manually at the eNB site [91].

**Antenna Height** Antenna height contributes to the determination of the coverage area and performance of the antenna due to its effect on the path loss and delay spread [52] (these terms will be defined more fully in Section 2.3.1.2). Generally the antenna height is only considered at the initial network planning stage as it is costly to change this once a network has been deployed.

**Transmission Power** The transmission power of the antenna determines its signal strength and coverage area. Altering the transmission power of the antenna is primarily used to compensate for the effects of pathloss and shadowing [77]. Increasing the transmission power of the antenna can be used to overcome some of these effects by increasing the range of the signal and therefore the size of the sector's coverage area, however it can also have a significant and negative effect on the interference experienced by neighbouring cells. The transmission power is commonly optimised with a number of constraints. For example a GA has been developed [78] to minimise the transmission power of base stations in a Code Division Multiple Access (CDMA) network while maintaining a sufficient bit rate to meet the QoS requirements of its UE.

## 2.3 Interference in LTE Networks

The main problem addressed by this research is interference in LTE networks. This section describes what interference is and how it arises in the context of an LTE network. Several existing approaches to mitigating interference are described.

### 2.3.1 Downlink Transmission in LTE Networks

In order to understand how interference arises in LTE networks it is important to understand how the eNBs and UEs communicate. This section therefore describes the downlink transmission scheme used by LTE networks in order to provide the context for the discussion of interference in Section 2.3.2 and of existing approaches to reducing interference in Section 2.4.

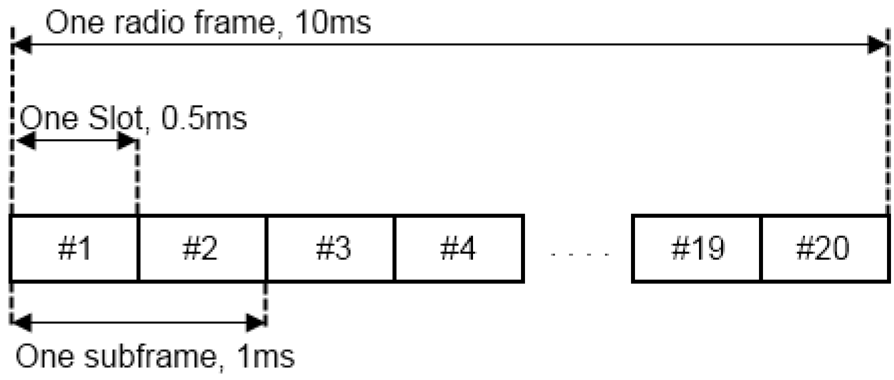


Figure 2.2: The time domain structure of LTE downlink physical resource.

#### 2.3.1.1 Downlink Resources

The radio resources which must be allocated by a eNBs to transmit data to UEs are frequency/time/space blocks. The spatial dimension is dependent on utilising multiple-antenna transmission and reception which is beyond the scope of this project.

The structure of the time-frequency resources is defined in the 3rd Generation Partnership Project (3GPP) standard documents [4, 15]. In the time domain the largest structure is a radio frame which has a duration of 10ms. This is further divided into 10 subframes each 1ms in duration, which can themselves be further divided into 2 slots each of 0.5ms in duration. Each slot can further be broken down into 7 Orthogonal Frequency Division Multiple Access (OFMDA) symbols. This can be more clearly seen in Figure 2.2.

The resources in the frequency domain are known as sub-carriers and are spaced at 15kHz each, as defined in [15]. These sub-carriers are aggregated into groups of 12, which for the duration of one slot are known as a Resource Block (RB) and occupy a total of 180kHz. See Figure 2.3 for a visual explanation of these divisions. The number of RBs available to each eNB is dependent on the range of frequency available to the network operator. See Table 2.1 for details.

Each RB can be subdivided into a number of Resource Elements (REs), typically 84 (12 sub-carriers $\times$ 7 OFMDA symbols) [15]. Some of these REs are reserved for specific purposes, for example reference signals, control signals and synchronisation signals which are not used to transmit data to the UE directly but rather are required

| Channel Bandwidth (MHz) | Number of RBs |
|---|---|
| 1.4 | 6 |
| 3 | 15 |
| 5 | 25 |
| 12 | 50 |
| 15 | 75 |
| 20 | 100 |

Table 2.1: The relationship between the operator's channel bandwidth and the number of available RBs [3]

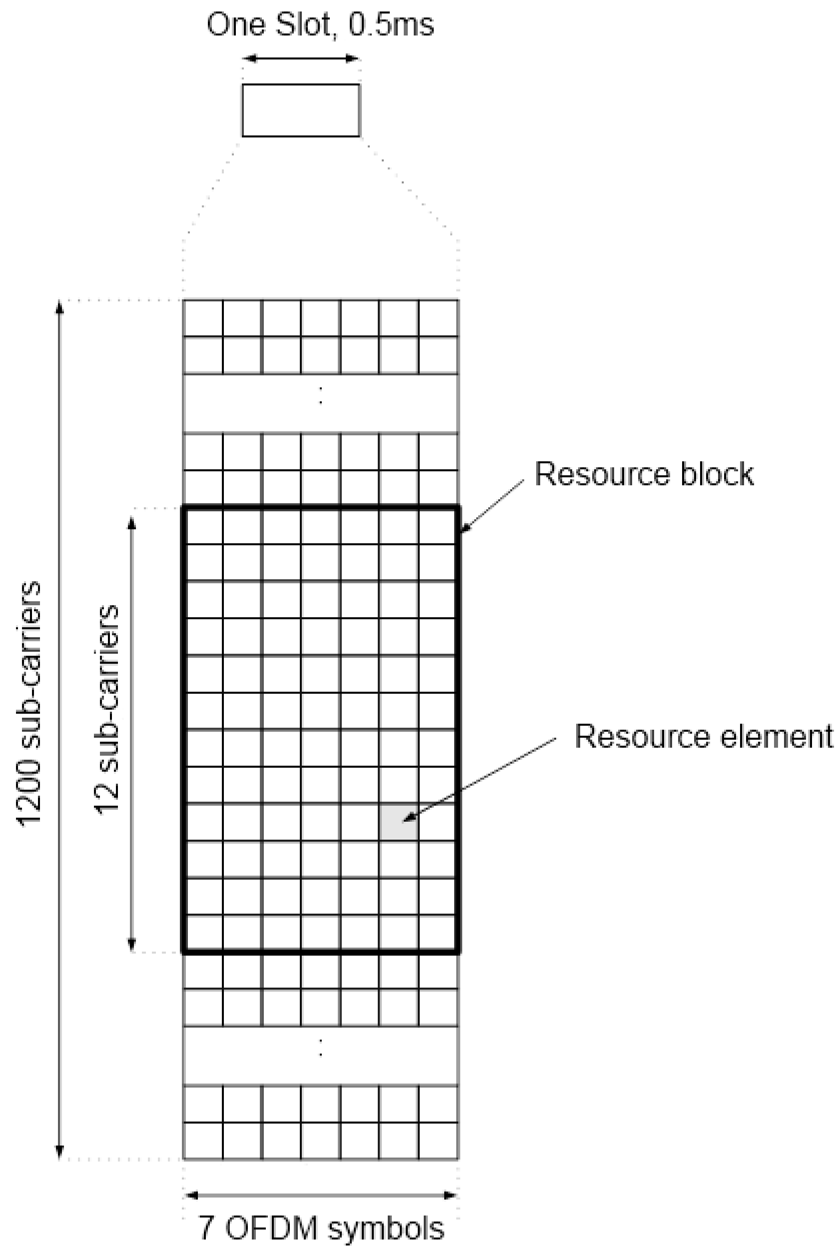


Figure 2.3: The frequency structure of the LTE downlink physical resource.

for the smooth running of the network [15]. The remainder of the REs are used to transmit data directly to the UEs.

The amount of information transmitted in each RE varies depending on the Modulation and Coding Scheme (MCS) used by the eNB to encode the data for transmission [17]. The modulation scheme used determines the amount of redundant data that is included in the transmission to aid the receiving UE in decoding it. This is dependent on the signal quality currently being experienced. This is discussed more fully in the following section. If the signal received by the UE is strong then more bits can be transmitted per symbol but if the UE is experiencing heavy interference or is otherwise experiencing poor signal quality then more redundant information is required.

#### 2.3.1.2 Signal Quality

The strength of the signal can be considered as its power. Each signal is transmitted by the eNB at a certain power level. The signal strength as experienced by a UE is determined by how much of that transmitted power it receives. In the following sections the various phenomena which lead to the dissipation of this power as it is transmitted are discussed.

Pathloss, or attenuation, is the reduction in power which is experienced by every electromagnetic wave as it travels through a medium such as air. Essentially the further the original signal travels from the transmitter the weaker it gets as the energy in it dissipates into the environment [82, 31]. Thus, aside from any other phenomena, as a UE moves further from its eNB, its signal will inevitably get weaker. Figure 2.4 shows the relationship between the pathloss experienced by a UE and its distance from the eNB.

Shadow fading, or simply shadowing, is the dissipation of power in a signal due to the effect of the environment through which it travels [61]. For example the strength of a wireless signal can be degraded due to being reflected by buildings or absorbed by trees. A UE which has a direct line of sight to its eNB will have significantly stronger signal than a UE which is behind a building.

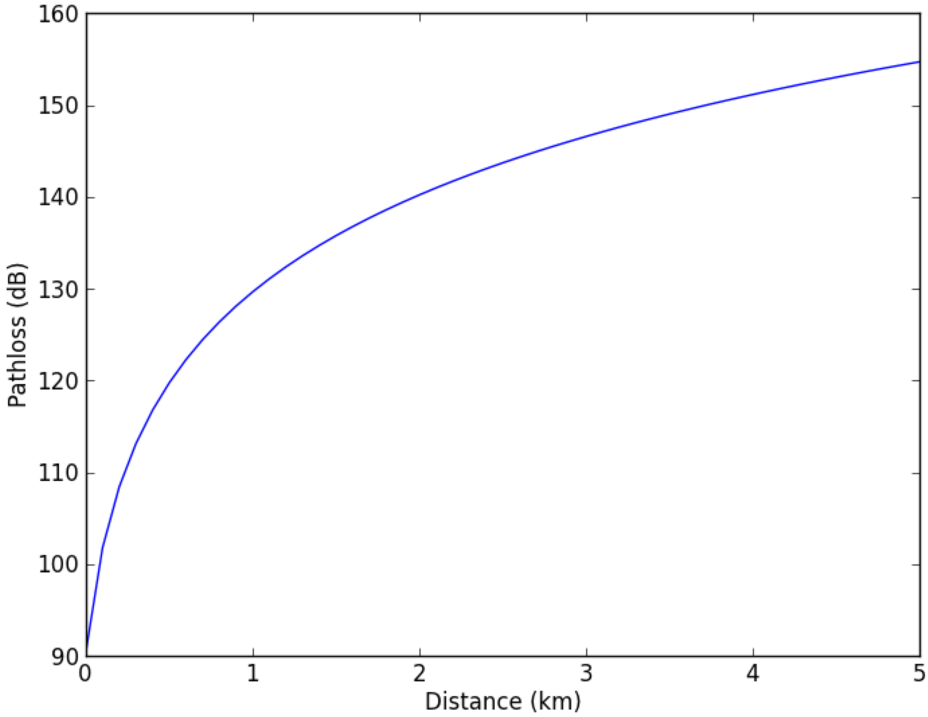


Figure 2.4: The effect of distance on the propagation loss experienced by a UE using the propagation model outlined in [6].

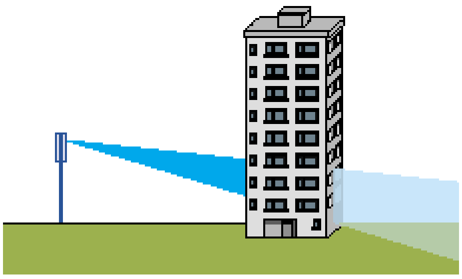

Figure 2.5: Shadowing causes signal strength to diminish due to environmental factors

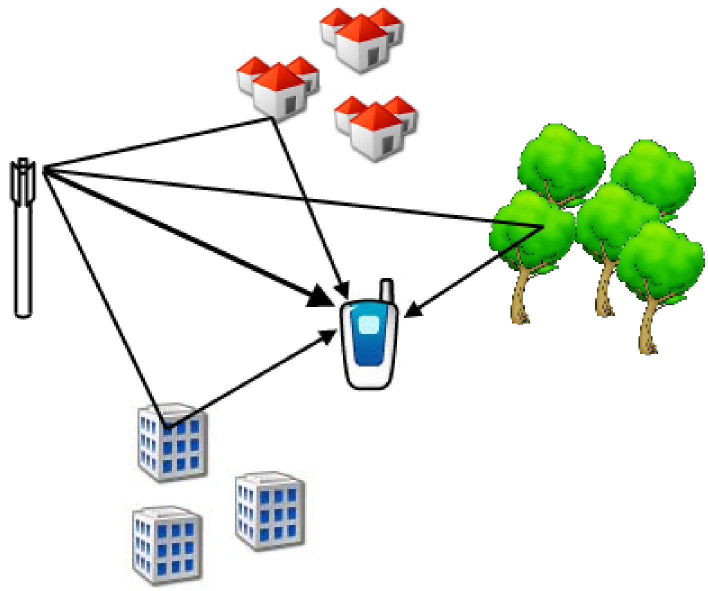

Figure 2.6: Multi-path fading causes multiple copies of the transmitted signal to be received by the UE due to interactions with the environment.

Another phenomenon which affects the strength of the received signal is known as multi-path fading [61]. This is caused by the reception at a UE of multiple copies of the same signal. These multiple versions of the original signal are caused when the signal is reflected or refracted by objects in the environment such as buildings or vegetation, see Figure 2.6. These reflections may arrive almost simultaneously, if the effect is due to local scattering, or after longer intervals due to multiple paths between the eNB and UE. As a UE moves the magnitude and phase of each reflected signal changes. This can result in either constructive or destructive interference, which can strengthen or weaken the signal.

#### 2.3.1.3 Scheduling of Resources

This section examines how RBs are allocated to UEs given that data is transmitted to UEs in RBs and a variety of factors affect the quality of the signal received over time. It is the job of the eNB's scheduler to determine which RB to allocate to which UE at any given time. An eNB maintains a data queue for each UE which it serves. Given other input data, such as the reported quality of a UE's channel or the current traffic load and priority, it must determine how to allocate the available

RBs in the most efficient manner.

As the eNB receives data to be transmitted to a UE, the data is stored in a buffer. This data is then divided into a number of RBs and transmitted to the UE at a given power level. The amount of data which can be transmitted to a UE on an RB is determined by the MCS employed by the eNB. This is determined dynamically by the eNB and is based on the signal quality experienced by the UE.

Thus, every 0.5ms an eNB will transmit zero or more RBs to each UE which it is serving. Typically the scheduler will attempt to satisfy the QoS constraints of as many UEs as possible by allocating the RBs and transmission powers of each sub-frame.

The eNB can transmit each RB at a different power level. This affects the quality of the signal received by the UE along with the interference experienced by other UEs

#### 2.3.1.4 CQI and Link Adaptation

In order to utilise the available resources most effectively eNBs attempt to match a suitable MCS to the signal quality received by the UE. This process is known as link adaptation [38]. In order to aid the eNB in this process the UEs can be configured to report the channel quality conditions (known as Channel Quality Indicators (CQIs)) to its serving eNB. This involves the UE measuring the strength of the signal it receives on a number of RBs and reporting it back to the eNB.

#### 2.3.1.5 Existing Schedulers

**Maximum Rate Scheduling** Maximum rate scheduling attempts to maximise the amount of data sent to UEs at the expense of fairness. This involves the eNB scheduling each RB to that UE which is currently experiencing the best channel conditions. This allows it to transmit the most amount of data possible per RB, thus maximising resources. However there is a clear cost in that UEs which are experiencing sub-optimal channel conditions may never be scheduled.

**Proportionate Fair Scheduling** Proportional Fair (PF) scheduling aims to provide a fairer approach to scheduling. It schedules a UE when its instantaneous channel quality is high relative to its own average channel condition over time.

### 2.3.2 Interference

To understand Inter-Cell Interference Coordination (ICIC) it is important to first understand how interference arises in LTE networks and how this affects performance. Each eNB serves a number of UEs within its coverage area. This area is determined by the transmission power of the eNB. This can be thought of as analogous to volume of the human voice, the louder a person talks the greater the area in which others can hear their voice.

When designing and deploying a network, one of the important tasks of a network operator is to determine where each eNB will be placed. In particular it is important that between eNBs there is an area where their coverage overlaps. This is to allow UEs to move between cells whilst still maintaining a single call or data session. For example in Figure 2.7 if UE X moves from cell A to cell B they will move through an area where there is no coverage and so their call will be dropped. In contrast Figure 2.8 shows that UE X can move freely between cell A and cell B when there is an area of overlap between them.

While this overlapping coverage is essential for allowing UEs to move seamlessly between cells it also gives rise to interference. Interference is experienced by a UE if, when receiving a transmission from its serving eNB on a given RB, it additionally receives transmissions on the same RB from one or more other eNBs. To illustrate Figure 2.9 represents two network cells, each of which is serving three UEs. Each cell has a total of 100 RBs available to serve these UEs and has split them into three groups each represented by a color. It is important to note that there is a small area of overlap between the cells. In this simple network setup UEs 1.B and 2.B will both experience interference. This is because they are both located in the area of coverage overlap and because they are being scheduled on the same RBs. In comparison, although they are being scheduled on the same group of RBs, UEs

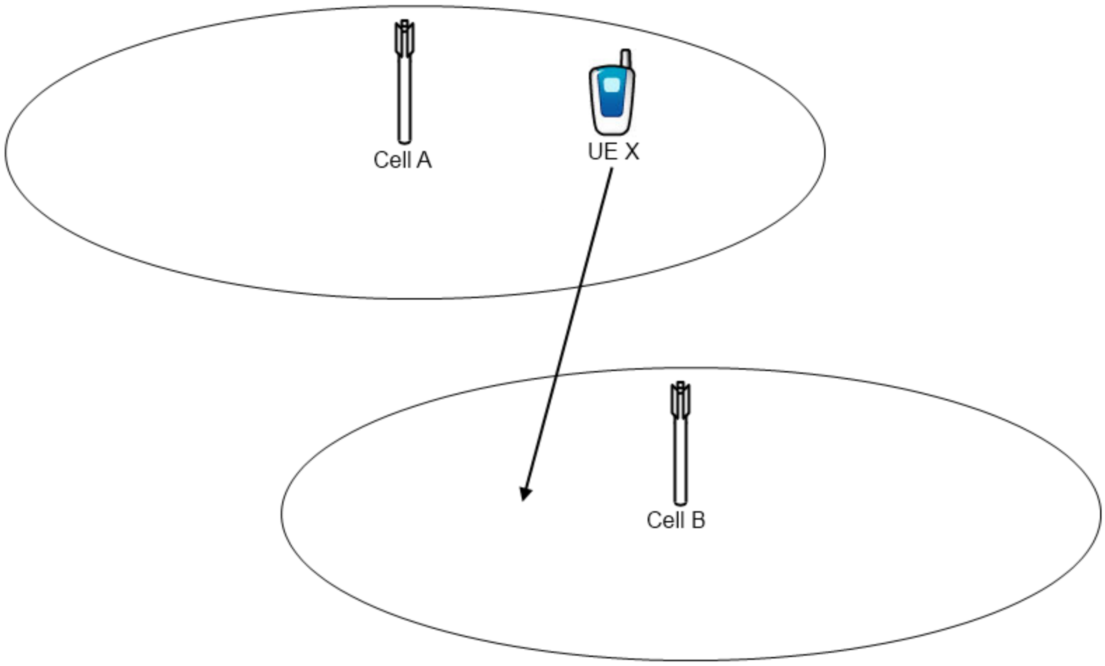


Figure 2.7: Network cells without overlapping coverage. The UE moving from cell A to cell B will lose its connection in the process.

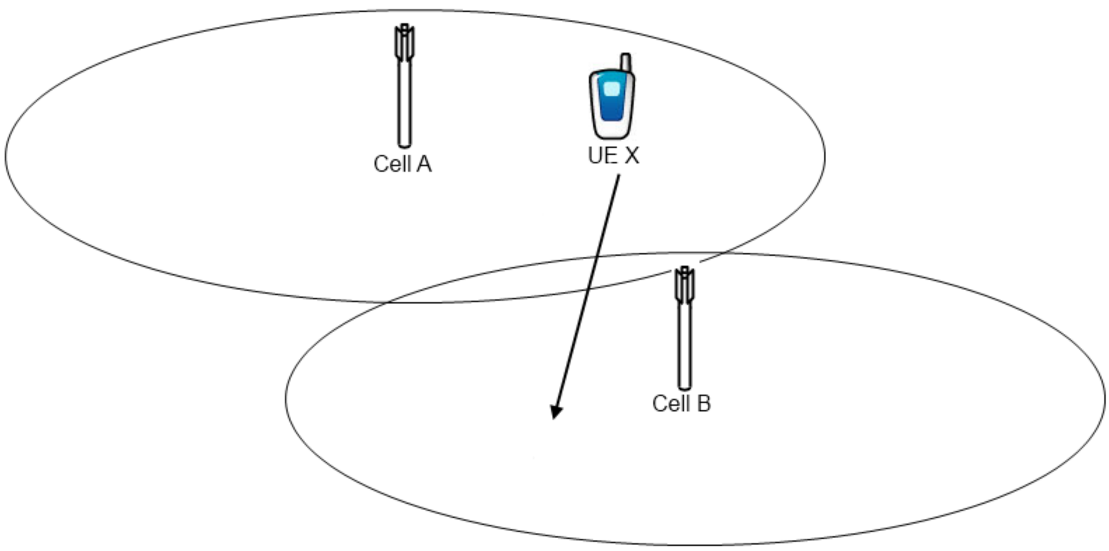


Figure 2.8: Network cells with overlapping coverage allowing the UE to move from one cell to the other without losing connection.

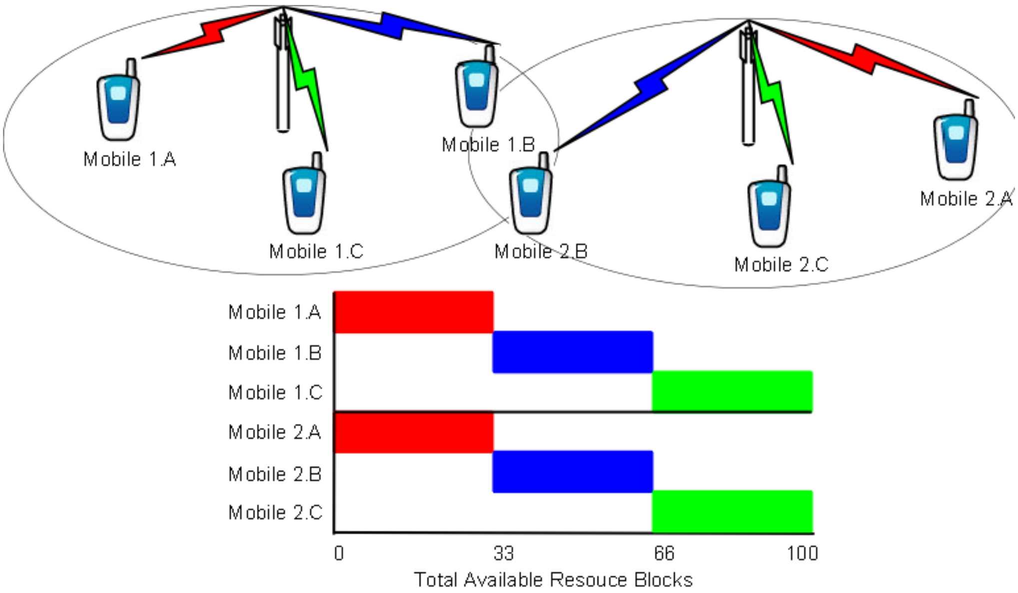


Figure 2.9: Mobile 1.B and 2.B will experience interference since they both occupy an area of overlapping coverage and are both being scheduled on the same group of RBs.

1.A and 2.A will not experience interference because they are not located in the area of coverage overlap and therefore will not receive the transmissions from the neighbouring eNB.

Interference is only experienced when the transmissions occur over the same RB. This is because RBs are orthogonal in the frequency domain and so adjacent RBs do not cause interference.

The primary effect of interference is to reduce the Signal to Interference plus Noise Ratio (SINR) of the UEs on the network, which has a direct effect on their throughput. The throughput of a UE can be modelled using the following equation [6]:

$$datarate(bps/Hz) = \begin{cases} 0 & if\ SINR < SINR_{min} \\ \alpha.S(SINR) & if\ SINR_{min} < SINR < SINR_{max} \\ datarate_{max} & if\ SINR > SINR_{max} \end{cases}$$

Where: $S(SINR)$ is the Shannon bound of the signal-to-interference-plus-noise ratio: $S(SINR) = log_2(1+SINR)$ bps/Hz. The $\alpha$ parameter, which represents implementation losses, takes the value 0.6, while the $SINR_{min}$ is -10dB and $SINR_{max}$

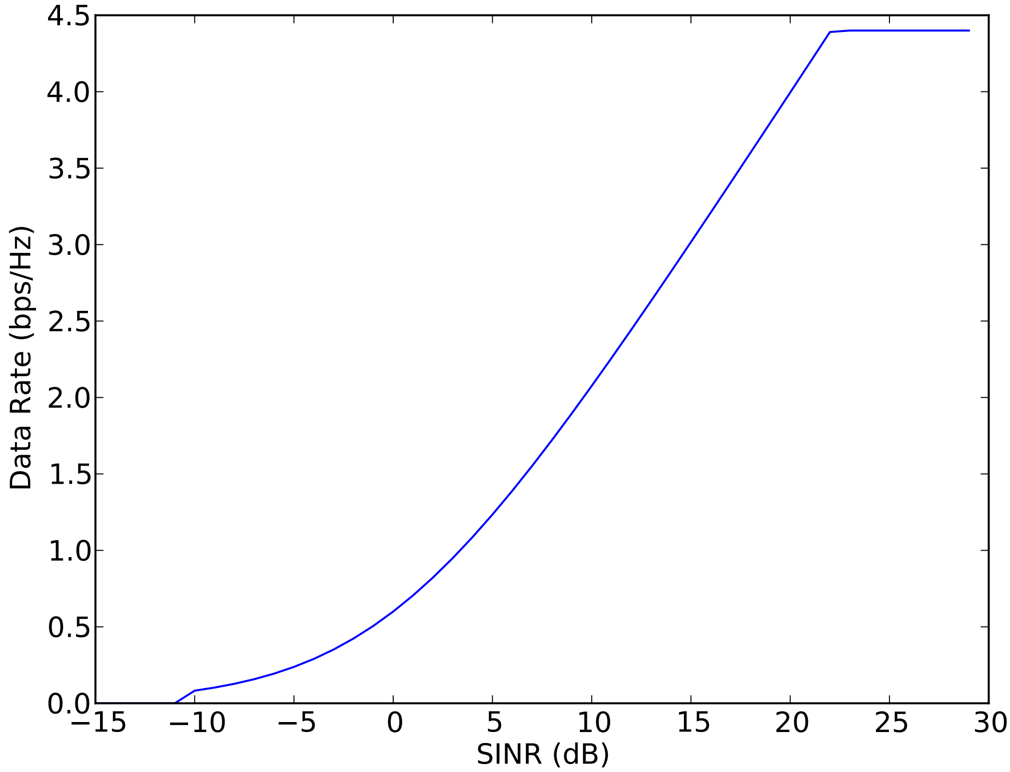


Figure 2.10: Relationship between SINR and UE datarate.

is 25db. The maximum datarate which can be achieved under this model is 4.4bps/Hz. Figure 2.10 shows the relationship between the SINR experienced by a UE and its maximum achievable data-rate.

The key component of this model is the SINR. This is the ratio of the strength of the signal received by the UE to the combined strength of the noise and interference received by the UE. It can be seen that an increase in the interference experienced by a UE causes the throughput of that UE to decrease.

## 2.4 Inter-Cell Interference Mitigation Techniques

The three main approaches to mitigating inter-cell interference which have been proposed [19] are described below.

### 2.4.1 Inter-cell Interference Randomisation

Inter-cell interference randomisation techniques incorporate some level of randomness to the transmission process in an effect to reduce the level of interference experienced by the network.

One approach that has been explored in the literature is that of cell-specific scrambling [47, 27]. In this approach a unique scrambling code is assigned to each eNB which is then used as part of the encoding of its transmissions. When these transmissions are received by a UE being served by that eNB the cell specific scrambling code is used to decode the transmission. As part of this process, provided the scrambling codes used are orthogonal, the interference from neighbouring cells will have been whitened or de-correlated. This refers to the fact that the power spectrum of the interfering signal will become more similar to that of white noise which has well understood statistical properties. This allows the UE to better differentiate the actual signal transmitted by its serving eNB from the interference transmitted by neighbouring eNBs.

An alternative approach to whitening the interfering signals which aims to improve the decoding performance of the system is known as Interleave Division Multiple Access (IDMA) [85]. A cell specific interleaving code is used by each eNB which achieves the same result as the cell specific scrambling code discussed above. The key improvement available in an IDMA system is that of iteratively coded multi-user detection. This allows the UE to improve their suppression of the interfering transmissions over time and thus can provide better performance than that of cell specific scrambling.

Another approach known as frequency hopping has also been proposed as a way to reduce the interference in a network [49]. It uses a randomised pattern to alternate the sub-carriers that are used by the UEs. Rather than continuing to schedule a UE on the same set of sub-carriers for the duration of its session frequency hopping dictates that the set of sub-carriers used by each UE change over time. Since a unique hopping pattern is assigned to each cell this means that the interference experienced will be averaged across all UEs rather than reducing the interference experienced by a specific set of UEs.

One basic scheme simply randomises the sub-carriers that are assigned to each UE [35]. Thus each UE transmission is randomly spread across the available time-frequency resources. Since this randomisation takes place at each eNB the interference experienced by UEs is spread randomly rather than being concentrated in a single section of the available time-frequency resources. This simple approach was shown to provide some performance improvement under a heavily loaded network.

Despite the research highlighted above it has been suggested that inter-cell interference randomisation is unlikely to provide the required improvement to the performance of cell edge UEs since it cannot provide any SINR gain to the UEs [20].

### 2.4.2 Inter-cell Interference Cancellation

Inter-cell interference cancellation is generally concerned with using antenna techniques to enhance the signal received by the UEs from the serving eNB and reduce the level of interference received from other eNBs.

One approach to inter-cell interference cancellation uses a combination of beam-forming and multiple antenna technology. Beam-forming allows the eNB to shape the overall antenna beam of its transmission in the direction of the receiving UE and can improve the signal strength received by the UE in proportion to the number of antenna used by the eNB. For small networks of 2 to 3 cells this has been shown to provide significant throughput gains [102].

Another approach to inter-cell interference cancellation which has been shown to reduce the impact of interference is the combined use of sub-carrier based virtual multiple-input multiple-output with a virtual signature randomiser [70]. Multiple-input multiple-output techniques can increase the throughput of an LTE network by using multiple antenna to transmit and receive the wireless signals. The multiple antenna can be used to increase the performance of the network in various ways such as pre-coding, spatial multiplexing or diversity coding, depending on the network environment. Virtual multiple-input multiple-output attempts to utilise the same concepts as multiple-input multiple-output but operates across the sub-carriers used to transmit the data rather than operating across multiple physical antenna. The virtual signature randomiser attempts to improve the UEs ability to distinguish the signals transmitted from its serving eNB from those transmitted from neighbouring eNBs. It does this by introducing an eNB specific random weight vector into the scrambling block included in the transmissions. This allows the UE to better differentiate the signals coming from the serving eNB from those originating from other eNBs.

### 2.4.3 Inter-cell Interference Coordination

ICIC seeks to apply restrictions to the downlink scheduling resources in a way which is coordinated with neighbouring cells. Typically this involves restricting which RBs are available or limiting the transmission power of certain RBs. These restrictions can actually reduce the level of interference in the system, rather than simply suppress the existing interference, but are more complex to implement.

Proposed ICIC schemes are classified as either static or semi-static schemes, depending on the level of communication between nodes required to coordinate the resources. Static schemes require no or at most very limited communication between eNBs, typically only at the initial setup phase. The coordination in these schemes comes primarily from the way the frequency is divided up when the network is initially set up. Semi-static ICIC schemes usually have more frequent communication between nodes in order to reconfigure the scheme depending on the network environment.

#### 2.4.3.1 Static Inter-Cell Interference Coordination Schemes

A static ICIC scheme proposed in [46, 48], known as Soft Frequency Reuse (SFR), divides the total available spectrum into a number of distinct groups known as sub-bands. These sub-bands are then allocated to the network's cells such that neighbouring cells are allocated bands which are orthogonal, thus minimising interference. The users at the centre of the cells can be served using the whole of the available spectrum, but at a lower transmission power which reduces the amount of

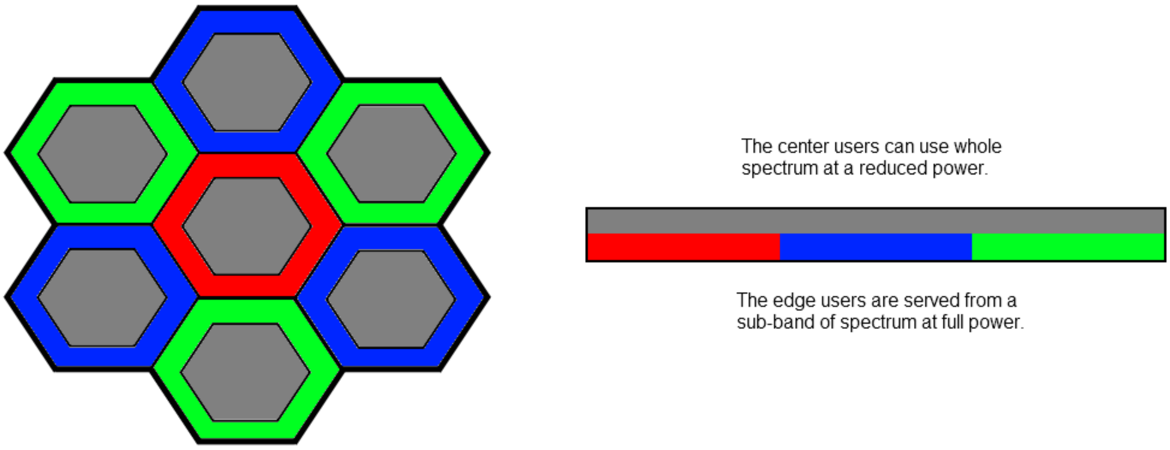


Figure 2.11: Frequency allocation in a static SFR scheme.

interference they cause to those UEs in neighbouring cells. The users at the cell edge are served at full power but can only be allocated resources within the sub-band of the cell.

Figure 2.11 illustrates this concept. It can be seen that the edge of each cell is only using a portion of the available spectrum while the centre of each cell is able to use the whole available spectrum. In the central cell the edge users can only make use of the red spectrum band, while the edge users of the neighbouring cells use the blue and green bands. This means that any transmissions to the edge users of the central cell will not cause interference to the edge users of the neighbouring cells.

This is a simple ICIC scheme which requires little overhead to implement and is therefore often used as the basis for more complex semi-static schemes. It does however limit the frequency utilisation of the edge users to 1/3 thus reducing their maximum throughput. The performance of this scheme will also depend greatly on the load of the network. A proposal by Alcatel [21, 22] improves the frequency utilisation of this method to 3/7 by sub-dividing each cell into 3 sectors.

#### 2.4.3.2 Semi-Static Inter-Cell Interference Coordination Schemes

In order to improve the flexibility of the static ICIC schemes, Siemens developed a semi-static scheme which builds on the previous proposals while allowing for greater flexibility depending on the network load [90, 88]. In this scheme the available spectrum is subdivided into 16 sub-bands which can be allocated in a flexible way between edge and centre users. The number of edge and centre users is then calculated dynamically and the sub-bands divided between them in a proportionate way. This allows the cells to alter their frequency allocation depending on the traffic load and user distribution.

Another scheme has been proposed which adapts to the ratio of edge to centre users along with the traffic demands of each group [99]. In this scheme the edge users are allocated a number of sub-bands depending on their load while the centre users can be allocated the entire available spectrum. Transmissions to the centre users are sent at a lower power level while those to the edge users are transmitted at full power. If there is heavy load in the edge area of one cell, it can borrow from the edge sub-bands of neighbouring cells provided they are under a much lighter load. This scheme provides greater spectral efficiency but does require signalling between nodes to facilitate the sharing of resources.

Softer Frequency Reuse (SerFR) is a scheme which builds on the traditional SFR scheme and utilises a modified PF scheduling algorithm [103]. In this scheme the entire available frequency spectrum is used by the whole cell. In each cell a sub-band of frequencies are transmitted at full power. These sub-bands are allocated to cells in an orthogonal manner. In order to determine which users are scheduled on which RBs, the scheduler calculates a priority value for each of the users. This priority value is based mainly on the current achievable data rate and the average data rate for that user, but is also modified by a $\gamma$ value. This $\gamma$ value depends on whether the RB being scheduled is part of the full power frequency sub-band and whether the user is an edge or centre user. These $\gamma$ values are configurable and so allow the algorithm to be flexible. For example certain $\gamma$ values would prevent any centre users being scheduled on full power RBs while others would not distinguish at all between edge and centre users.

#### 2.4.3.3 De-centralised Inter-Cell Interference Coordination Schemes

The ICIC schemes discussed above can be described as centralised, that is the parameters are set as part of the network planning process and are not adjusted dynamically. De-centralised schemes by contrast are designed to be more adaptive to the network environment and as such can alter their behaviour without any central control. For example one approach to de-centralised ICIC is described in [93]. In this scheme the available RBs are split into 3 non-overlapping sub-groups each of which is assigned to a separate cell. When an eNB requires extra resources it can request the use of one of the RBs from its neighbouring eNBs by using the X2 link. The average throughput achieved by this scheme is reported as being slightly higher than that of SFR however the throughput of the cell edge UEs remains similar to SFR. It is worth noting however that the use of the X2 link was independent of the scheduler used, in this case Round Robin, which should allow the scheme to be used with any scheduling algorithm.

## 2.5 Evolutionary Algorithms

In this thesis extensive use of EAs, in the form of GAs, will be made to find optimal solutions to various problems. Optimisation problems are concerned with finding the optimal solution to a given problem. Many interesting problems, such as the travelling salesman problem, can be formulated as optimisation problems. Optimisation problems themselves can also be seen as a search problem where a search space containing all possible solutions must be searched for the optimal solution. This firstly assumes that there exists some function, called the objective function, which can ascertain the quality, or fitness, of a solution.

The combination of all possible solutions and the objective function can be visualised as a graph, like that in Figure 2.12, known as a fitness landscape. Here the y-axis represents the fitness, as determined by the objective function, and the x-axis represents all possible solutions. This graph contains a number of local maxima and a single global maximum. The purpose of an optimisation algorithm is to search this space for the solution that represents the global maximum. However the more complex the landscape the more difficult this task proves. The two optimisation algorithms used in this project are introduced in the following sections.

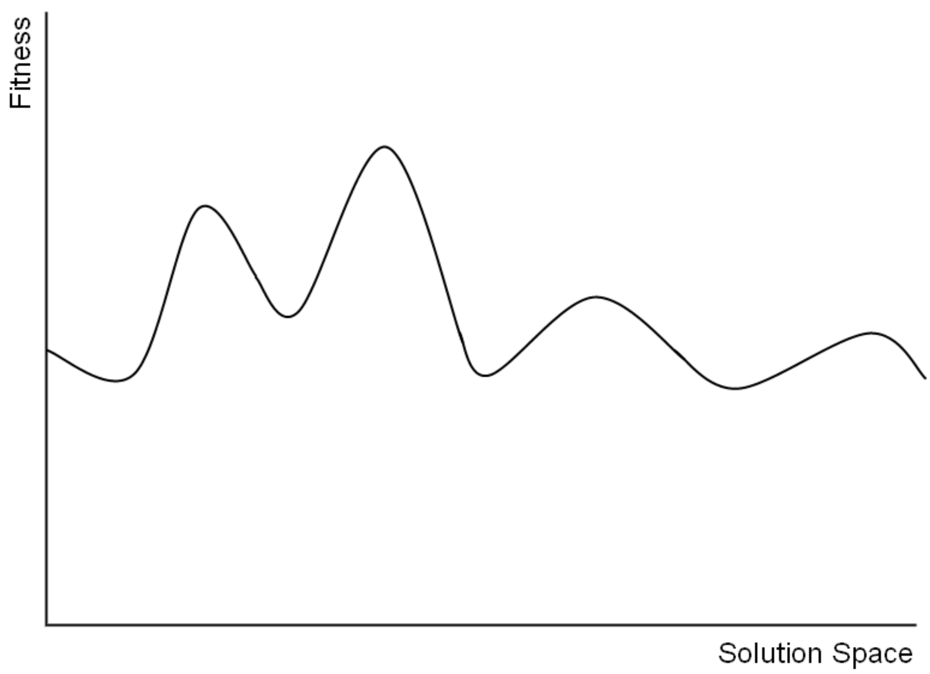


Figure 2.12: An example one-dimensional fitness landscape.

### 2.5.1 Single-Objective Genetic Algorithms

Traditional GAs [76, 57, 86, 58] offer a robust approach to search and optimisation problems inspired by genetics and natural selection. This is achieved by maintaining a population of possible solutions to the given problem. The population is initialised by creating a number of randomly generated solutions. During each iteration a new population is created by selecting solutions from the existing population as parents and recombining them to produce offspring using crossover operators. These offspring are then mutated with a given probability and added to the new population. This process is depicted in Figure 2.13.

GAs tend to be more robust and less likely to become trapped in a local maximum than more basic search algorithms such as hill-climbing, in part due to the diversity gained by maintaining a number of different solutions.

In order to optimise a problem a GA must be given a means to represent solutions for the problem as a chromosome. This is known as the encoding of the problem. Often this takes the form of a list of values where each value represents some variable in the solution. This list is known as a chromosome and each value is known as a gene. Each chromosome is then encoded as a list of values such that the list

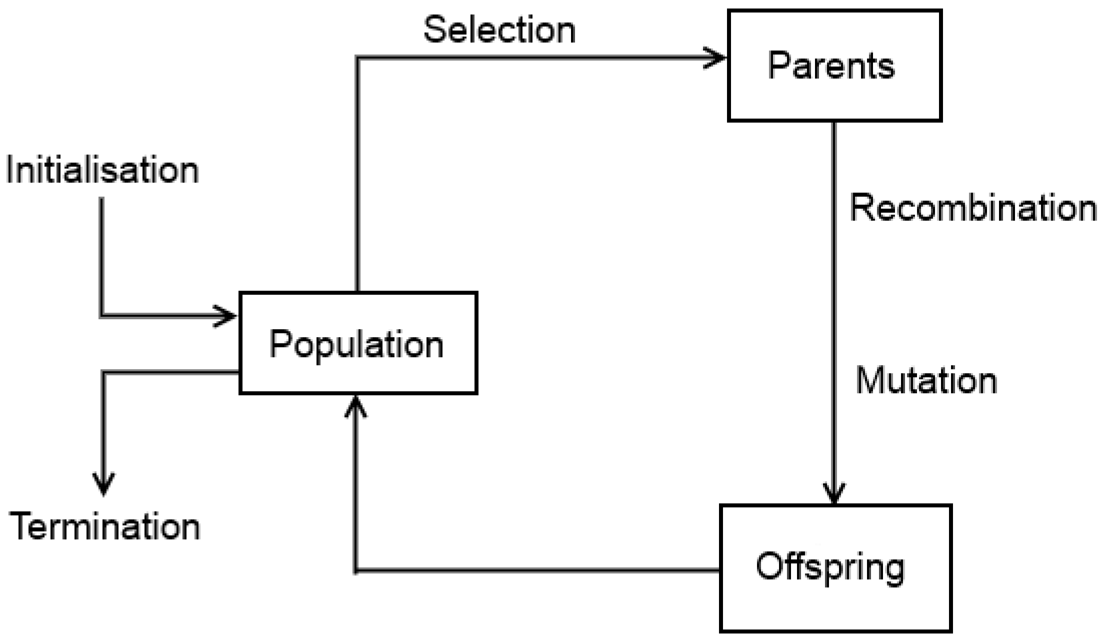


Figure 2.13: Overview of the main steps taken by a GA during each iteration.

represents completely the solution to the problem. For example if a solution to some problem could be represented by five binary digits then a randomly generated chromosome could be represented as follows (Figure 2.14):

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 0 |

Figure 2.14: Structure of a GA chromosome.

There are a number of operators available for GAs to utilise. Operators are the means by which the GAs create random solutions from their existing populations. The main types of operator are mutation operators, crossover operators and selection mechanisms.

**Mutation Operators** The mutation operator produces a single new child chromosome from an existing parent chromosome. There are a number of possible operators which can be used for this purpose [28], however a commonly used approach is simply to select a random gene and choose a new random value for it. This approach is known as "single point mutation" and is based on biological point mutation [63].

Given our example chromosome illustrated above such a mutation would operate as seen in Figure 2.15.

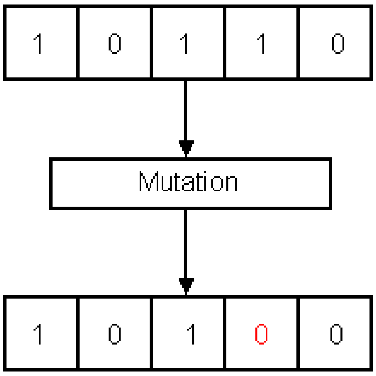


Figure 2.15: Mutation on a GA chromosome.

**Crossover Operators** The crossover operator produces two or more child chromosomes from two or more parents. As with mutation operators, there are a number of possible operators which can be used [29], several of which are briefly outlined below.

Single point crossover makes use of a single, randomly chosen, mark point between the first and the last gene of a chromosome. To construct a child chromosome, genes from index 0 to the mark point are taken from the first parent and the remaining genes are taken from the second parent. This can easily be used to generate two children as seen in Figure 2.16.

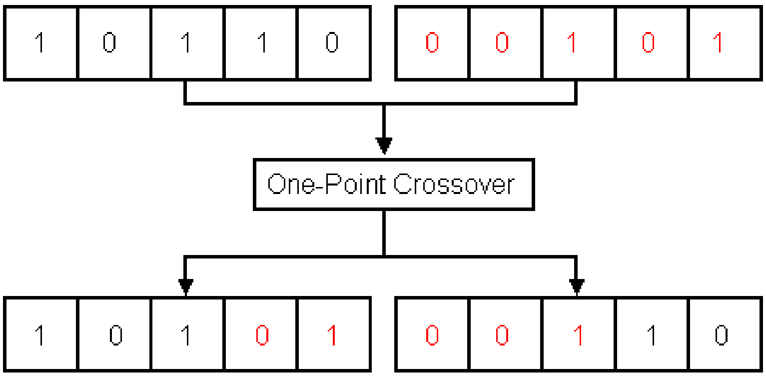


Figure 2.16: One point crossover operating on two GA chromosomes.

Two point crossover operates in a similar manner except that two points are chosen randomly. To construct a child chromosome genes from index 0 to the first point are taken from the first parent, genes from the first point to the second point are taken from the second parent with the remaining genes being taken again from the first parent. This is illustrated in Figure 2.17.

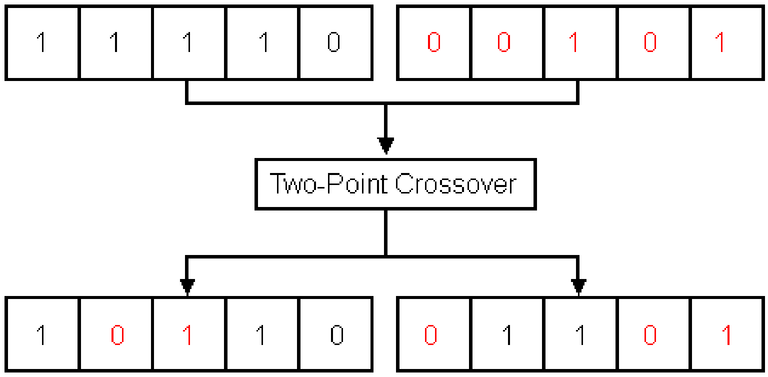


Figure 2.17: Two point crossover operating on two GA chromosomes.

Uniform crossover operates slightly differently. A child chromosome is created by iterating over each gene of the parents. With a given probability (usually 50%) the gene is chosen from the first parent or else it is taken from the second. This is illustrated in Figure 2.18.

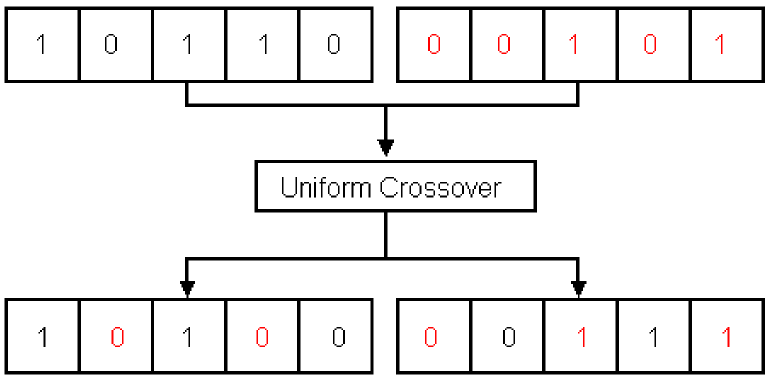


Figure 2.18: The operation of uniform crossover on a GA chromosome.

There is no clear consensus regarding which of the crossover operators is most effective [32]. The contribution that each operator makes to the performance of a GA can be understood in terms of the schemata theorem and the building block hypothesis.

The schemata theorem [64] is an attempt to explain the problem solving power of GAs in terms of schemata. Schemata are particular gene patterns which may be contained in a candidate solution. The binary coded candidate solution "1101", for example, contains the schemata "11##", "1#01", "#1##" and "11#1" where the

"#" symbol matches any value. The order of a schemata is the number of non-# digits it contains and the defining length is the distance from the outermost non-# digits in the schemata. It assumed that the fitness of a candidate solution is directly tied to its schemata such that fitter candidate solutions must contain better schemata. Since these fitter candidate solutions contribute more of their genes to the next generation more good schemata are passed on to the next generation improving its fitness. The theorem states that shorter, fitter schemata increase with subsequent generations. The purpose of crossover operators is to combine parents in such a way as to combine the good schemata in each where possible.

Each crossover operator has a different effect on the propagation or destruction of the schemata present in a candidate solution [32]. Propagated schemata are those that are passed on to the child during crossover whereas schemata that experience disruption are not passed on. Under uniform crossover, schemata of all orders are equally likely to be disrupted. Under single point crossover and two point crossover however the defining length of a schemata is directly proportional to its likelihood of destruction. So the choice of crossover operator determines the likelihood of schemata with certain characteristics being passed on to subsequent generations.

The building block hypothesis [59] states that the power of GAs is in their ability to find good building blocks. These are low order schemata with a short defining length which complement each other when present in a candidate solution. The formation of these building blocks can be encouraged by the use of a coding scheme that meets two criteria. Firstly related genes are located close together in the chromosome and secondly there is little interaction between genes. This interaction between individual genes is known as epistasis and can cause a reduction in the effectiveness of crossover operators. For example is has been found that the performance of two point crossover drops if these coding scheme recommendations are not met [33] while uniform crossover still performs well.

**Selection Mechanisms** The selection mechanism determines which of the parents in a population will be selected as parents in the production of the next generation. One common approach to selection is known as fitness proportional or roulette wheel selection [30]. In this approach the probability of a chromosome being selected is directly proportional to their fitness. The fitter a chromosome is the more likely it is to be chosen for mutation or crossover. This ensures that in general the fitter solutions remain in the population while at the same time there is some level of diversity, in that some less fit solutions are also likely to be included.

Another common approach is known as tournament selection [74]. This involves running several tournaments between randomly chosen chromosomes. The tournament size determines the number of chromosomes that are randomly selected to be entered into the tournament pool. The fittest of the chromosomes within the tournament pool are then selected.

A key factor in the choice of selection mechanisms is the trade-off that exists between selection pressure and genetic diversity [34, 60]. Selection pressure is the tendency to select the best members of the current generation to propagate to the next generation while genetic diversity is required to ensure that the search space is adequately explored. There is a tension between these two factors since a higher selection pressure will likely lead to a decrease in the genetic diversity of the population.

Tournament size allows for adjustment of selection pressure by adjusting the tournament size. The larger the tournament size the more likely that the fitter candidate solutions will be selected and thus the greater the selection pressure. Larger tournament sizes however are also associated with a loss of genetic diversity in the population. In contrast the selection pressure and genetic diversity resulting from use of the proportional selection mechanism are dependent on the fitness function itself allowing for less transparent understanding of its performance.

### 2.5.2 Multi-Objective Genetic Algorithms

A traditional GA can only search for a single optimal solution, which in many cases is all that is required. That is, a traditional GA can only use a single fitness measure or objective function. However in more complex cases, such as those under investigation in this thesis, there are a number of possible fitness measures or objective functions which must all be optimised. Often these are even competing measures such that an improvement in one measure will only be obtained at the cost of another measure. In these cases a more advanced version of the GA, known as a Multi-Objective Genetic Algorithm (MO-GA), can be used.

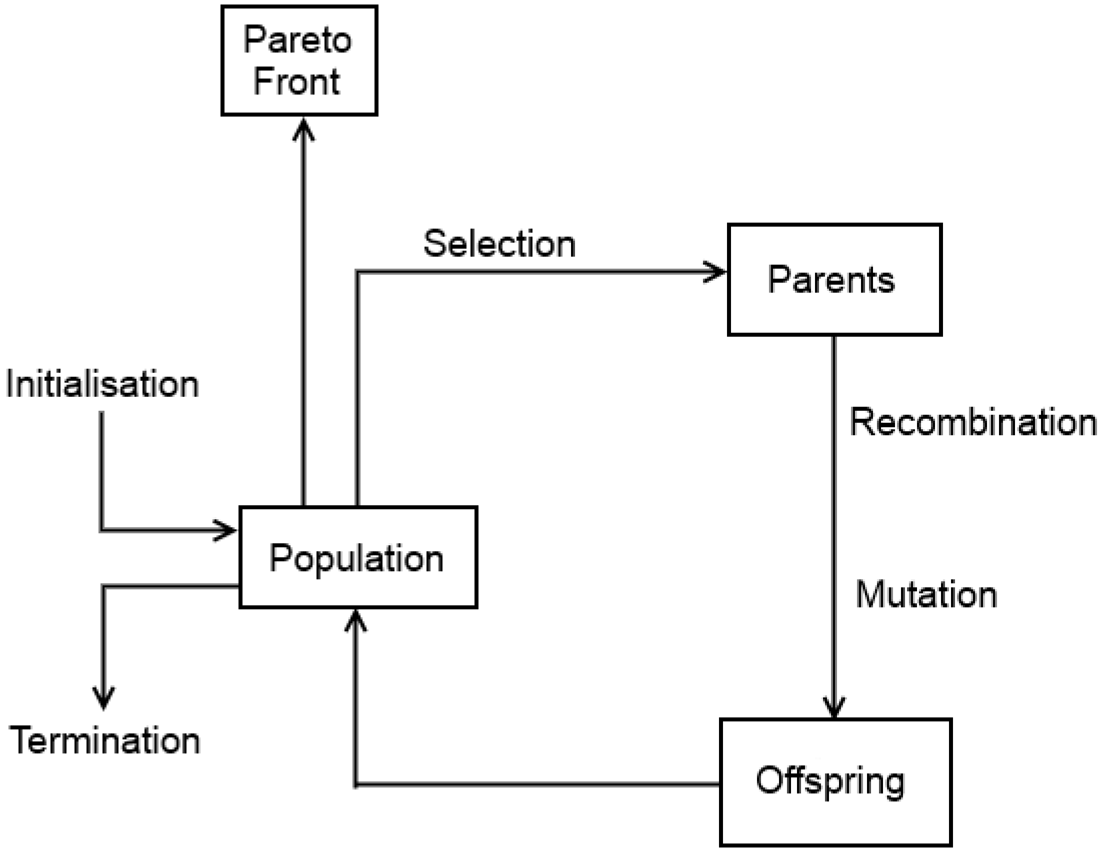


Figure 2.19: Main steps taken by a Multi-Objective Genetic Algorithm

An MO-GA operates in much the same way as a traditional GA (see Figure 2.19) but with the addition of a Pareto front. The Pareto front is a set where the non-dominated candidate solutions discovered by the algorithm are retained. A non-dominated candidate solution is one where no other candidate solution achieves a better fitness across all objectives. A candidate solution is said to dominate another solution if it is considered fitter according to all objective functions.

When plotted the candidate solutions contained in the Pareto front often result in a graph like that in Figure 2.20 where there are a number of solutions which perform well. In this case some of the candidate solutions perform better in fitness measure × than y and there are a number in between. The end user can then choose which candidate solution is optimal depending on the priority they assign to each fitness measure.

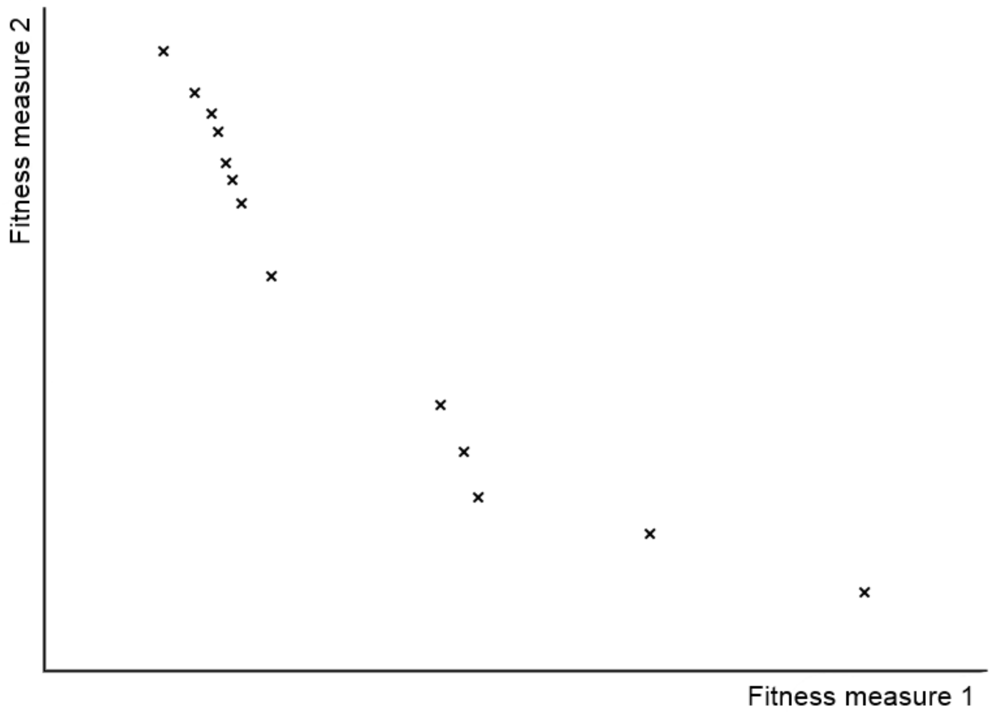


Figure 2.20: An example of a Pareto front.

To more fully describe the operation of an MO-GA the standard NSGA-II [41] algorithm will be discussed before highlighting the differences between it and the algorithm used in this research. The acronym NSGA-II stands for Non-dominated Sorting Genetic Algorithm. A diagrammatic overview of its operation can be seen in Figure 2.21.

Initially the NSGA-II algorithm creates a random population of candidate solutions $P_0$ of size $N$ which are then sorted by non-domination level. Solutions with a non-domination level of zero are known as Pareto Optimal solutions since they are not dominated by any other candidate solutions. An initial offspring population $Q_0$ of size $N$ is then produced using the standard crossover and mutation operators. Once this setup phase of the algorithm is complete the general procedure for a $t$th generation continues as follows. First a combined population $R_t = P_t \cup Q_t$ is constructed. This combined population is now of size $2N$. The candidate solutions

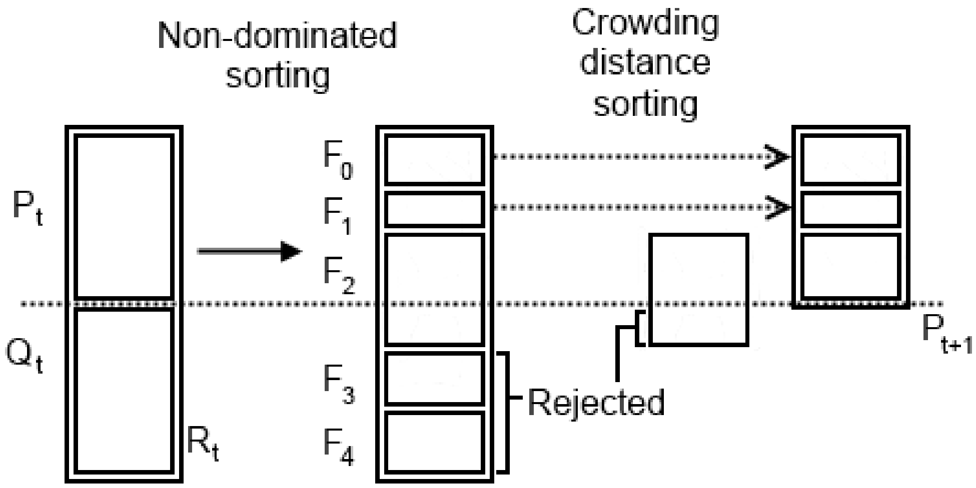


Figure 2.21: Overview of NSGA-II algorithm.

contained in $R_t$ are then sorted using a fast, non-dominating sorting function which sorts the candidate solutions into a number of fronts according to their domination count. Those solutions which are not dominated by any other solution are sorted into front $F_0$, while those that are dominated by only a single other solution appear in front $F_1$.

Beginning at front $F_0$ each front is added to the next generation's population $P_{t+1}$ until it has size $N$. If a given front is too big to fit into the new population its members are sorted by the crowded comparison operator and the highest ranked solutions are chosen for inclusion. The crowded comparison function prefers candidate solutions which are in a less crowded region of the front which helps to maintain a diverse set of candidate solutions in the new population.

Once the new population has been populated with $N$ candidate solutions it is used to produce a new offspring population $Q_{t+1}$ by using the standard crossover and mutation operators. Binary tournament selection is used for parent selection with the selection criterion being the crowded comparison operator. The new population $P_{t+1}$ and offspring population $Q_{t+1}$ are then used as the input to the next generation.

The algorithm used as part of this research shares similarities from the NSGA-II algorithm described above but also takes inspiration from other approaches such as the Pareto Archived Evolutionary Strategy described in [69]. A graphical overview of its operation can be seen in Figure 2.22. As with NSGA-II an initial population

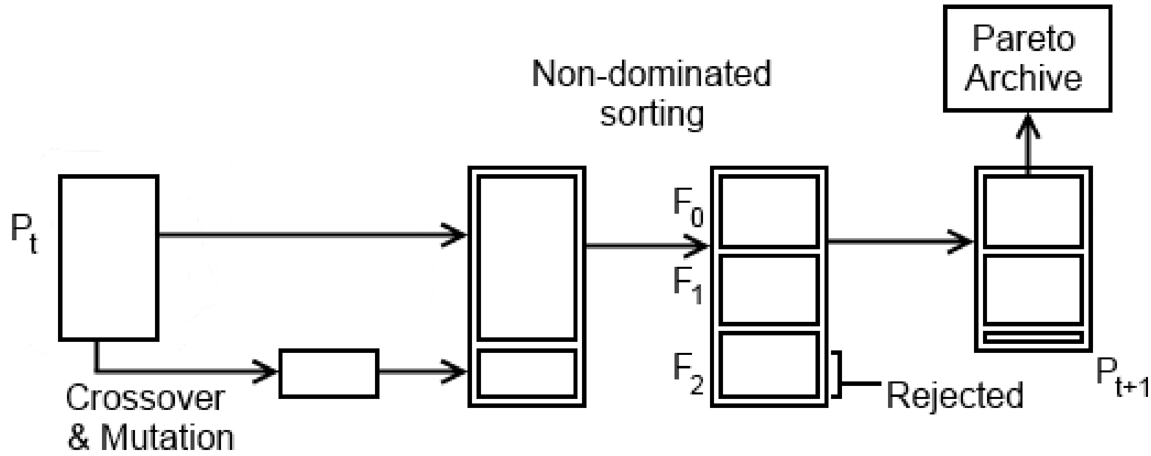


Figure 2.22: Overview of the MO-GA used in this research.

of size $N$ is created. During each generation two candidate solutions are selected for crossover using fitness proportional selection along with another for mutation. This results in three new candidate solutions being added to the existing population. The domination counts for each solution are then calculated and any non-dominated solutions are added to a separate Pareto front archive. Since the size of the current population is $N + 3$ selection is again used to select those candidate solutions which will survive to the next generation. This algorithm lacks the crowding distance sorting that is used in the NSGA-II algorithm and is therefore likely to lead to less diverse solutions over time.

### 2.5.3 Evolutionary Algorithms in Telecommunication

EAs have been widely used within the telecommunications research community.

For example antennae play an increasingly key role in wireless networks. The design of antennae is a challenging problem since the antennae themselves must generally be cheap and as simple as possible to implement at the hardware level but must satisfy certain electrical requirements. For example NASA have used EAs to develop antennae specifically for communication with satellites [55]. The resulting designs not only improved on the performance of those antennae designed in a conventional fashion but the speed of producing the designs was both considerably faster and cheaper. Parallel EA were show-cased by Villegas et al. [95] as a robust and fast way to produce designs for a patch antenna to be used in wireless networks. EAs have also been used to optimise the design of multiple-antenna arrays [67] that are becoming increasingly important as LTE adoption continues.

Within wireless networking in particular there are several areas where EAs have proved be a valuable optimisation approach. The placement of eNBs (or equivalent) within a wireless network is an important design decision. It affects not only the cost of infrastructure for the network operator but also the area covered by the network and the quality of the service provided to the users. A number of EAs have been developed to tackle this problem both in its general form [36] and in its application to specific network technologies such as 3G [72]. Related to this is the problem of configuring the antennae on each eNB in an optimal fashion. As previously discussed there are a number of parameters which can have a significant effect on the performance of the network such as the transmission power and the antenna downtilt and beam-width [101].

Within the specific area of scheduling and ICIC in LTE networks EAs have been developed that optimise the uplink scheduling of UEs. The centralised scheme presented in [92] uses an EA to uplink scheduling of UEs in LTE networks which make use of multiple antenna technology (known as Multiple-Input Multiple-Output (MIMO)). It was found that EAs could provide almost optimal results.

Another EA has been developed to optimise the scheduling of frequency resources in LTE networks with a particular focus on the frequency of channel quality reports which are sent from the UEs to the eNB [98]. In LTE the UEs can report back the channel quality they experience on each RB to their serving eNB. This allows the eNB to adapt its transmission encoding scheme to the channel conditions that each UE experiences. The higher signalling rate that is required is costly and so it is important to optimise the reporting frequency to maximise the opportunity of the eNB to encode transmissions (and as such improve the data rate of the UE) while at the same time minimising the signalling cost by minimising the number of reports sent.

Another EA has been developed to optimise LTE networks using semi-smart antennae [100]. Semi-smart antennae allow the amplitude and phase of their transmissions to be altered which allows the coverage pattern produced to be varied thus allowing the size and shape of each cell to be optimised. The semi-smart antennae were optimised using an EA with two key objectives: to provide coverage for all users, and to maximise the total throughput of those users. Since the size and shape of the cell's coverage area was being optimised it was possible for a network to be generated which had coverage gaps. This is undesirable since any user in a coverage gap would not be able to send or receive any traffic and thus their throughput would effectively be zero. Since the coverage patterns were being optimised it was also important to optimise the transmission power simultaneously in order to minimise the interference experienced by the users and thus maximise their throughput.

These highly constrained problems illustrate how GAs can be used to optimise several parameters simultaneously within a set of constraints.

## 2.6 Conclusion

This chapter has described aspects of the downlink transmission of LTE networks, in particular those aspects that give rise to the problem of interference. A number of approaches to mitigating interference were discussed with a specific focus on ICIC. Some existing ICIC schemes presented in the literature were described and contrasted. These schemes will be referenced in the following chapters as a number of alternative approaches to ICIC are developed and introduced which are influenced by them. EAs and their existing applications in the wireless networks have been introduced and discussed as these techniques will be incorporated into the new ICIC schemes presented in the subsequent chapters.

# Chapter 3

# Localised Organisation at the Level of a Single eNodeB

In Chapter 2 the problem of interference in Long Term Evolution (LTE) networks was introduced along with the concept of an Inter-Cell Interference Coordination (ICIC) scheme to mitigate the effects of interference. Two existing ICIC schemes, Soft Frequency Reuse (SFR) and Softer Frequency Reuse (SerFR), were also described. The main theme of both these existing approaches was to divide the available frequency into a number of bands which could then be assigned to User Equipments (UEs) using different power levels. In each of these approaches a new scheduler was developed that was able to take advantage of the different power levels available to schedule the bands in a way that reduced the interference and improved the performance of the network.

This chapter describes a new ICIC scheme which is based on the SerFR described in the previous chapter. A parameterised scheduler is presented which can be optimised by an Genetic Algorithm (GA) to allow it to be adapted to any network environment. We refer to this as a local optimisation in that the scheduler has no knowledge of the neighbouring Evolved NodeBs (eNBs).

A number of experiments are described detailing the performance of this new algorithm and comparing it with the existing ones. A network simulator is also described that was developed in order to fairly compare the new algorithm with those in the literature.

## 3.1 Adaptive Softer Frequency Reuse

The Adaptive Softer Frequency Reuse (aSerFR) is based on the SerFR scheme discussed in Section 2.4.3.2. As with the SerFR scheme the available frequency is split into a number of bands some of which are transmitted at full power and the remainder at a reduced power level. This allows the central UEs to be scheduled on the reduced power Resource Blocks (RBs) thus reducing the interference that they cause in neighbouring cells. The cell edge UEs can then be scheduled on the full power RBs in order to increase their throughput.

SerFR uses a proportional fairness scheduling algorithm to schedule UEs to RBs. To determine which UE to schedule on a given RB a priority value, $P_{k,j}(n)$, is calculated for each UE on that RB. The UE with the maximum priority value is scheduled on that RB. The priority value is defined as

$$P_{k,j}(n) = \frac{DRC_{k,j}(n)}{R_k(n)}\gamma_{k,j}, \tag{3.1}$$

where $DRC_{k,j}(n)$ is the achievable data rate the the $k^{th}$ UE on the $j^{th}$ RB at time $n$ and $R_k(n)$ is the low-pass filtered averaged data rate of the $k^{th}$ UE at time $n$. $R_k(n)$ is defined as

$$R_k(n+1) = (1-\alpha)R_k(n) + \alpha DRC'_k(n) \tag{3.2}$$

where $\alpha$ is the forgetting factor and $DRC'_k(n)$ is the average data rate of the $k^{th}$ UE at time $n$. The $\gamma_{k,j}$ in equation 3.1 is used to modify the priority value depending on on whether the $k^{th}$ UE is at the centre or edge of the cell and whether the $j^{th}$ RB is a high or low power RB. The value of $\gamma_{k,j}$ can be adjusted to ensure that despite having lower average data rates UEs on the cell edge are more likely to get scheduled on high power RBs.

The aSerFR scheme utilises the same proportional fair scheduling algorithm discussed above but modifies its behaviour based on a number of parameters. This allows the aSerFR scheduler to be optimised by an GA for use in particular network environments. The optimisation is performed by changing a number of parameters each of which are described below.

As with SerFR UEs are classified as either edge or centre users. This classification of UEs is based on a signal quality threshold which is one of the parameters which can be optimised by the GA. Since edge users typically experience greater interference from neighbouring cells it is a safe assumption that those users with the poorest signal quality are edge users, see [50, 40, 54]. This signal quality threshold ranged from -15dB to +15dB and as such was encoded using real value encoding. To mutate this value uniform mutation was used to generate a uniform random value between the upper and lower bounds of the desired range.

The eNB will transmit at full power to edge users and at a reduced power to central users. The level of reduced power which is used for the central users is another parameter of the scheduler which can be optimised by the GA. The edge users are transmitted to using 100% of the available power while central users will be transmitted with only some proportion of that depending on the parameter value set by the GA. This reduced power factor was represented in the chromosome using a real value between 1.0, representing full power, and 0.0 representing no power and as such was encoded using real value encoding. Uniform mutation was again used to generate new uniform random values for this gene during mutation.

In a similar fashion to SerFR a certain number of the available RBs are transmitted using full power and the remainder are transmitted at a reduced power level. The number of RBs which are transmitted at full power is also a configurable parameter in this scheduler. It is expected that the number of RBs required by the edge users will vary depending on the traffic conditions. The proportion of full power RBs is represented as a real value ranging between 1.0, representing the state that all RBs are transmitted a full power, and 0.0, representing the state that none of the RBs are transmitted at full power. As with the previous real value encoded parameters uniform mutation was used to generate a uniform random value for mutation.

The scheduling mechanism adopted by aSerFR is the same Proportional Fair (PF) scheduler used in the SerFR scheme, with one modification. On occasion the scheduler will schedule its users randomly rather than using the PF scheme.

| Short-name | Description | Values |
|---|---|---|
| *RandomTrigger* | The chance of the scheduler operating randomly | 0.0 - 1.0 |
| *ProportionFullPowerRBs* | The proportion of RBs to transmit at full power | 0.0 - 1.0 |
| *ReducedPowerFactor* | The proportion of full power to use when transmitting to central users | 0.0 - 1.0 |
| *EdgeThreshold* | The threshold at which UEs are considered edge users | -15dB - 15dB |
| *RBAllocationScheme* | The scheme used for determining which RBs will be transmitted at full power | *Orthogonal* or *Random* |

Table 3.1: The GA parameters for the aSerFR scheduler.

The probability of this random scheduling occurring is governed by a parameter which will be optimised by the GA. The probability of this random behaviour being triggered was modelled as a real value ranging from 1.0, representing continuous random scheduling, and 0.0, representing no random scheduling at all. As with the previous real values uniform mutation was used to generate new random values for this parameter.

The other difference between the SerFR scheme and the aSerFR scheme is the orthogonality of the full power RBs. The SerFR scheme assigns these in a strictly orthogonal manner such that neighbouring cells transmit at full power on different RBs. The aSerFR supports the orthogonality of the full power RBs but also allows the option of randomly distributing these such that there is a possibility that neighbouring cells will transmit at full power on some of the same RBs. The orthogonality of the RB allocation scheme is represented in the chromosome as a single bit where 1 represents the full power RBs being allocated in a strictly orthogonal manner and 0 represents them being allocated by random distribution. When this gene was mutated the bit was simply flipped.

These five parameters govern the behaviour of the aSerFR scheduler. They are listed in Table 3.1 for easy reference, along with their possible values, and are referenced using the shortened form shown for the remainder of this document.

## 3.2 LTE Network Simulator

In order to model the behaviour of each of the ICIC schemes and allow for the development of new ones an LTE network simulator was built. The source code for the simulator is available at `http://downthewire.co.uk/projects/jLTE-Simulator/`. The design and operation of the simulator are described in the follow sections.

### 3.2.1 Overview of Simulator Execution

The simulator is designed to simulate specific aspects of the Radio Access Network (RAN) of an LTE network. In particular it is mainly concerned with the calculation of interference, data rates and the scheduling of RBs to UEs. In addition the simulator is only concerned with the downlink direction and therefore consideration of the uplink direction is outside the scope of this research. This is due to two considerations. The first is the additional computational cost of calculating the interference in the uplink direction. The computation cost of calculating the interference in either direction is directly related to the number of entities which may interfere with each other. In the downlink direction there are only 19 eNBs whose signals may cause interference, while in the uplink direction there can be as many as 1150 UEs causing interference. This represents a significant increase in the computational cost of calculating the uplink interference. The second factor in focusing on the downlink is its greater relative importance to network users. Downlink speeds are typically more important in the applications which network users value. For example streaming video and online gaming can require significant downlink speeds and there is growing demand for these services, whereas there is very little demand for uplink intensive services.

The network to be simulated consists of 19 eNBs each with 3 sectors as recommended by 3rd Generation Partnership Project (3GPP) [5]. A number of UEs are then randomly distributed across the network area. The specific number of UEs generated can be varied in order to simulate networks under different levels of user density.

At a high level the simulator simply runs the following algorithm:

- initialise the network
  - setup the network elements: eNBs, sectors, UEs
  - determine which sector is to serve which UE
- for each iteration:
  - calculate the downlink power between each sector and each UE
  - schedule the RBs to UEs using the given scheduling algorithm
  - calculate the signal power and interference experienced by each UE
  - calculate the datarate of each UE

**Initialisation** A number of initialisation steps must take place before the simulation can begin. The most important aspect of this is the creation of a channel between each UE and each sector. This channel models the quality of the radio channel between a UE and a sector by calculating a number of things such as the pathloss experienced by the signal between the sector and the UE, the shadowing experienced, and the location of the UE with respect to the sector. In order to determine which sector should serve a given UE the channel between the UE and each sector is calculated and the channel with the best downlink gain is chosen.

The pathloss, which also takes shadowing effects into account, is modelled using the propagation model detailed in Section 4.5.2 of [6].

### 3.2.2 Datarate Model

The datarate model used by the simulator to calculate the datarate of individual UEs is defined in [6]. Using this model the datarate of a given UE can be calculated from its Signal to Interference plus Noise Ratio (SINR) using the following equation:

$$datarate(bps/Hz) = \begin{cases} 0 & if\ SINR < SINR_{min} \\ \alpha.S(SINR) & if\ SINR_{min} < SINR < SINR_{max} \\ datarate_{max} & if\ SINR > SINR_{max} \end{cases}$$

Where: $S(SINR)$ is the Shannon bound: $S(SINR) = log_2(1 + SINR)$ bps/Hz. The $\alpha$ parameter, which represents implementation losses, takes the value 0.6, while the $SINR_{min}$ is -10dB and $SINR_{max}$ is 25db. The maximum datarate which can be achieved under this model is 4.4bps/Hz.

Using this model we can determine the maximum datarate that can be achieved in our simulator. The total available bandwidth in our simulation is 18,000,000 Hz (100 RBs $\times$ 180,000 Hz per RB). Since the maximum datarate is 4.4bps/Hz under this model, the maximum total datarate is 75.5 Mbps (4.4 bps/Hz $\times$ 18,000,000 Hz).

### 3.2.3 Channel Estimation Model

In order to model the wireless channel between the UEs and eNBs to a reasonable degree of accuracy it is necessary to implement a channel model. There are two main types of channel model which can be used:

- empirical channel models
- statistical channel models

Empirical channel models are based on empirical measurements taken in a variety of wireless propagation environments. They are considerably more detailed than statistical models as they calculate precisely a large number of paths which a transmission may take along with details such as the angle of departure, angle of arrival and phase change of each individual path from the eNB to the UE.

While empirical models do provide a more realistic approach to simulating wireless channels the level of detail of the calculations required to utilise them in our simulator proved to be slow for practical purposes.

Statistical channel models are generally simpler and are therefore more useful for simulations. The simulator makes use of a statistical model know as the Rayleigh Fading model which has been found to approximate closely the channel capacity in urban areas [37, 9, 89, 87, 43, 56].

Due to the computation load in calculating the gain values resulting from the channel gain model these are produced in batch before any simulations are run. This reduces the time required to run the simulations since the fading values do not need to be calculated each time a simulation is run.

The speed of the UEs mainly effects the fading values generated by the channel model. The faster a UE is moving the faster their signal quality will vary. For example see Figure 3.1.

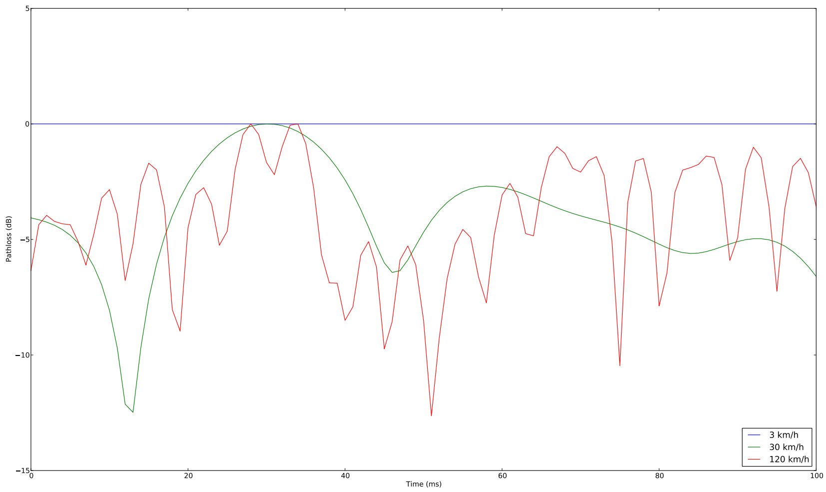


Figure 3.1: The effect of UE speed on the variation of signal strength.

### 3.2.4 Traffic Model

The UEs modelled by the simulator are categorised into four traffic groups: Light, Medium, Heavy and Full. The category a UE is placed in determines both the probability that there is traffic to be transmitted to it each iteration but also the number of RBs which are to be scheduled for that UE. The specific values for these parameters can be found in Table 3.2.

| Traffic Category | Probability of Data | Max RBs to transmit |
|---|---|---|
| Light | 20% | 5 |
| Medium | 50% | 15 |
| Heavy | 80% | 100 |
| Full | 100% | 100 |

Table 3.2: The probabilities associated with the traffic classes.

The number of UEs grouped into each class is then determined by the overall traffic profile used by the simulator. There are four profiles: Light, Mixed, Heavy and Full. The proportion of UEs in each traffic class is shown in Table 3.3.

This traffic model is not based on any empirical data about traffic loads but was instead designed as an approximation of the various levels of load that could be placed on the resources of the network. From the perspective of the ICIC problem the particular type of traffic, whether voice data or packet data, and its latency requirement are not of interest. Rather the number of RBs required to serve the traffic is the most important consideration. This traffic model captures the fact that, from the network's perspective, less demanding UEs will require less RBs to be served while more demanding UEs will require more RBs to be served. This is reflected in the Full traffic mode which is analogous to the the full buffer traffic model found in the literature [26, 22, 6, 103].

| Traffic Category | Proportion of Light UEs | Proportion of Medium UEs | Proportion of Heavy UEs | Proportion of Full UEs |
|---|---|---|---|---|
| Light | 70% | 20% | 10% | 0% |
| Mixed | 20% | 60% | 20% | 0% |
| Heavy | 0% | 0% | 100% | 0% |
| Full | 0% | 0% | 0% | 100% |

Table 3.3: The proportion of UEs created in each traffic class.

The aim of this traffic model is to allow for a fuller understanding of the performance of the scheduling algorithms by seeing how their performance changes with network load. As such this traffic model is not designed to be a realistic model of user behaviour. Rather it is intended to be a simple heuristic approach to modelling the level of load placed on the the network and therefore the level of contention for the radio resources. For example the heaviest possible load is when all the UEs on the network constantly need to be scheduled which is represented by the Full traffic level. In this case there is maximum contention for the network's radio resources and the scheduling algorithms will be operating under the most constrained environment. The other levels reflect network environments in which the contention for resources is decreasing.

### 3.2.5 Performance Metrics

During its operation the simulator gathers detailed statistics about the performance of the network. These metrics are used to evaluate and compare the relative performance of each of the ICIC algorithms that run on the network.

The most important metrics for this research are the average UE throughput and the 5th percentile UE throughput. These are the measurements typically used in the literature to evaluate ICIC schemes [39, 103, 48, 40]. It is important to note that both of these metrics are calculated using the average throughput of the UEs during the course of the simulation. The 5th percentile UE throughput is obtained by sorting the UEs by their average throughput. The average is then taken of the average throughput of the worst performing 5% of UEs. In the literature the 5th percentile UE throughput is used as a proxy for measuring the performance of the cell edge UEs. It is assumed that those UEs which experienced the worst throughput are those that are most affected by interference which are most likely to be those UEs at the cell edge [50, 40, 54].

It is important to note that these two metrics are competing constraints. This is due to the fact that there are a finite number of RBs which can be scheduled during each iteration and since the amount of data transmitted across each RB varies with the network conditions of the UEs to which it is scheduled.

As the 5th percentile UE throughput increases it is at the expense of the average UE throughput. In order to increase the 5th percentile throughput more RBs must be assigned to the cell edge UEs. These RBs will transmit less information to those UEs than they would to cell centre UEs due to the Modulation and Coding Scheme (MCS). Thus to increase the throughput of the edge UEs necessarily involves decreasing the throughput of the central UEs.

### 3.2.6 Validation

Unfortunately there are no standardised benchmarks against which to compare our simulator. Therefore in order to verify that our simulator behaves as expected a number of simple test cases were created and the output of the simulator compared to that of the theoretical output.

In order to verify the implementation of our simulator a number of tests were run. For example it was noted earlier that the maximum datarate that could be achieved with the simulator was 75.5 Mbps. To verify that the simulator was indeed capable of achieving this datarate a simple network setup was created which consisted of a single eNB and a single UE. This provided near optimal network conditions and allowed the UE to achieve the maximum datarate. This confirmed that the model implemented by the simulator was behaving as expected.

The other main verification step taken was the comparison of the performance of the baseline schedulers. While a direct comparison using the absolute performance was impossible (since there are no standardised simulator implementation, or even standardised simulation parameters) a relative comparison was done. Regardless of the implementation of the simulator the relationship between the performance of several of the baseline schedulers is well understood and should be observable. For instance the maximum rate scheduler should always achieve the best average UE throughput but will always perform very poorly in terms of 5th percentile UE throughput. The SFR and SerFR schedulers should generally outperform the other schedulers in terms of 5th percentile UE throughput since they are designed to optimise this metric [103, 46, 48].

## 3.3 Baseline LTE Schedulers

These existing schedulers are described in the literature and are described briefly here. They are used to compare against the results from the aSerFR scheduler.

### 3.3.1 Random Scheduler

The Random scheduler is a very simple scheduler which is designed to show the effect of scheduling when channel conditions are not taken into account by the scheduler. The schedulers used as part of the ICIC schemes discussed previously all make use of UE feedback regarding the channel conditions to inform their scheduling decisions. The random scheduler does not take into account such information when scheduling

RBs to UEs. It is used in the experiments in this and the following chapters as a benchmark against which to compare those schedulers which are sensitive to the channel conditions experienced by the UEs.

### 3.3.2 Round Robin Scheduler

Round robin scheduling [39] assigns the available RBs in a fair though simplistic way. It is fair in that the same number of RBs are made available to all the UEs. However due to the fact that it does not take into account the variations in channel quality it can also be considered an unfair way of scheduling since it will not provide the same service quality to all the UEs. This is because of the fact that those UEs which experience better signal quality will be able to use their RB more efficiently thus enjoying a better service quality than that those UEs which experience degraded signal quality.

Since round robin scheduling does not take advantage of the channel quality conditions it will lead to lower system performance. It does however guarantee that no UEs are starved of resources since all UEs will get an equal share.

### 3.3.3 Maximum Rate Scheduler

Maximum rate scheduling (also known as Max-C/I scheduling) [39] is a type of scheduling which takes into account the channel conditions being experienced by each of the UEs. Since the channel quality experienced by UEs within a cell typically varies independently due to effects like multi-path fading (fading caused by the reception of multiple copies of the same signal at the UE) there is generally at least one UE whose channel quality is near its peak. By scheduling the UE with the best channel quality the eNB maximises its throughput because the UEs with the best channel quality can make the most efficient use of the radio resources.

Maximum rate scheduling however cannot be considered fair in many circumstances. Whether maximum rate scheduling can be considered fair depends primarily on the causes of the channel quality variation. If all UEs have a similar channel quality which varies only due to effects like fast fading or multi-path fading then they will experience a very similar average throughput. However if the variations in channel quality are due to factors such as distance, shadow fading or fading due to speed then it is very unlikely that maximum rate scheduling will behave fairly. If there are UEs whose average channel quality is always less than others then they will never be scheduled and will essentially be starved of radio resources resulting in zero throughput.

### 3.3.4 Proportionate Fair Scheduler

PF [39, 65] scheduling attempts to find a balance between the approaches taken by maximum rate scheduling and round robin scheduling. Maximum rate scheduling makes the most efficient use of the radio resources available at the expense of the performance of UEs with poor channel conditions while round robin is scrupulously fair in its allocation of radio resources but does not deliver equal throughput to all UEs. PF scheduling makes use of the channel quality information available while also being fair in its distribution of radio resources. It does this by scheduling the UEs with the relatively best channel quality conditions. It does this by monitoring the channel conditions of each UE over a particular time window. When making a scheduling decision the relative channel quality for each UE is calculated by comparing the average channel quality of that UE to its current channel quality. This approach means that even those UEs who experience significantly worse channel conditions in general will receive radio resources when their channel quality peaks relative to its own average.

### 3.3.5 Soft Frequency Reuse Scheduler

As discussed in Section 2.4.3.1 SFR is an ICIC scheme that attempts to reduce interference in neighbouring cells by dividing up the available spectrum into a number of sub-bands which are served with different power levels. The full power sub-bands are reserved for use by the cell-edge UEs while the remaining reduced power sub-bands are used to serve the cell-centre UEs. These sub-bands are allocated to cells on an orthogonal basis such that no neighbouring cell will have the same sub-bands assigned to its cell-edge UEs.

While no specific scheduler is discussed as part of this ICIC scheme a modified PF scheduling algorithm has been proposed [103] which takes into account the status of the UE being scheduling, whether it is a cell-edge or cell-centre UE, and which sub-band it should be scheduled on.

### 3.3.6 Softer Frequency Reuse Scheduler

As discussed in Section 2.4.3.2 SerFR is an ICIC scheme which is built on the SFR scheme. Like SFR, it divides the available frequency in sub-bands some transmitted with full power and some with a reduced power factor; unlike SFR it is more flexible in the allocation of these sub-bands to UEs.

The scheduler used by the SerFR scheme is also a modified PF algorithm which takes into account the status of the UE as either cell-edge or cell-centre and also the power level of the sub-band.

## 3.4 Experiments

These experiments were designed to compare the performance of the optimised aSerFR scheduler, developed as part of this research, with that of the baseline schedulers described in Section 3.3.

In order to compare each of the algorithms thoroughly the experiment was run on a number of different network environments. In practice this meant a different number of UEs (see Table 3.4), different movement speeds of the UEs (see Table 3.5) and also different traffic patterns of the UEs (see Table 3.6). An experiment was run for each different combination of these parameters, resulting in 27 different network environments.

The number of UEs in each network environment was chosen to provide a reasonable number of UEs per cell. The simulated network was composed of 19 cells (see Table 3.7) and so 285 UEs represent lightly loaded cells with an average of only 15 UEs per cell, whereas the network environment with 1150 UEs represents a network where the cells are very heavily loaded. The traffic levels represent both the probability and amount of data arriving at an eNB for a given UE as described in Section 3.2.4. The UE speeds were chosen based on the standard simulation parameters set out by 3GPP [5].

| Number of UEs | Average number of UEs per cell |
|---|---|
| 285 | 15 |
| 680 | 35 |
| 1150 | 60 |

Table 3.4: The possible number of UEs in each network environment and the resulting average number of UEs per cell

| Traffic Levels |
|---|
| Light |
| Mixed |
| Heavy |
| Full |

Table 3.5: The possible traffic levels of the UEs in each network environment

| Speed (km/h) |
|---|
| 3 |
| 30 |
| 120 |

Table 3.6: The possible speeds of the UEs in each network environment

In order to evaluate the performance of the baseline algorithms each of them was run ten times for each network environment with different random seeds. The results for each of these runs were then averaged and plotted in the graphs seen in Section 3.5.

To evaluate the performance of the aSerFR scheduler a Multi-Objective Genetic Algorithm (MO-GA) was run in order to optimise its parameters for each network environment. The MO-GA was designed to optimise two metrics: 5th percentile throughput, and average throughput. The MO-GA produces a number of results which together form a Pareto front based on these two metrics. The solutions produced can all be considered equal in terms of fitness although they will have different characteristics in terms of their 5th percentile and average throughput values. This makes explicit the trade-offs that must be made between 5th percentile throughput and average throughput since they are competing constraints.

The simulator described above was used to evaluate the chromosomes produced by the MO-GA. Due to the time required to run the MO-GA (approximately 5 days) it was only run once for each network environment. However, in order to ensure that the simulator produced a representative result for any given chromosome, each chromosome was evaluated 10 times using a different random seed each time. These 10 values were then averaged to give the fitness value for that chromosome which was used by the MO-GA.

### 3.4.1 Simulation Parameters

The simulation parameters shown in Table 3.7 are mainly taken from [39, 5].

| Parameter | Value |
|---|---|
| Number of eNBs | 19 |
| Number of UEs | 252, 680, 1150 |
| UE speed | 3kmh, 30kmh, 120kmh |
| Data generation | Light, Mixed, Heavy Traffic profiles |
| Distance attenuation | $L = 128.1 + 37.6 \times log(distance)$ [6] |
| Shadow fading | Log normal, 8dB standard deviation |
| Multi-path fading | Rayleigh fading |
| Cell layout | Hexagonal grid, 3-sector sites, 57 sectors in total |
| Cell Radius | 500m inter-site distance |
| Simulation Time | 500 ms |

Table 3.7: The parameters used for the simulator in experiment 1.

The layout of the 19 cell network can be seen in Figure 3.2.

Note that the simulation run time shown in Table 3.7 was arrived at after a pilot study had been conducted to determine the required number of iterations which were needed to achieve a stable and accurate result. Each of the baseline algorithms was run under a particular network environment (1150 UEs, full traffic, 120km/h) for 2500 iterations (2500 ms of network time).

As can be seen from Figures 3.3 and 3.4 while the average throughput did not stabilise after 500 iterations in all cases, the 5th percentile throughput did reach a stable state after 500 iterations. Since the 5th percentile throughput is the more important measure for determining the effectiveness of an ICIC solution and since it would have been prohibitive to run the simulation for longer, 500 iterations was chosen as a reasonable simulation time.

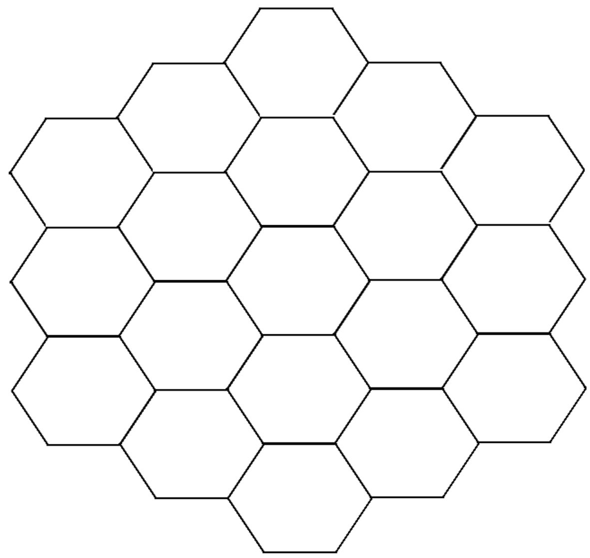

Figure 3.2: The 19 cell network used in experiment 1.

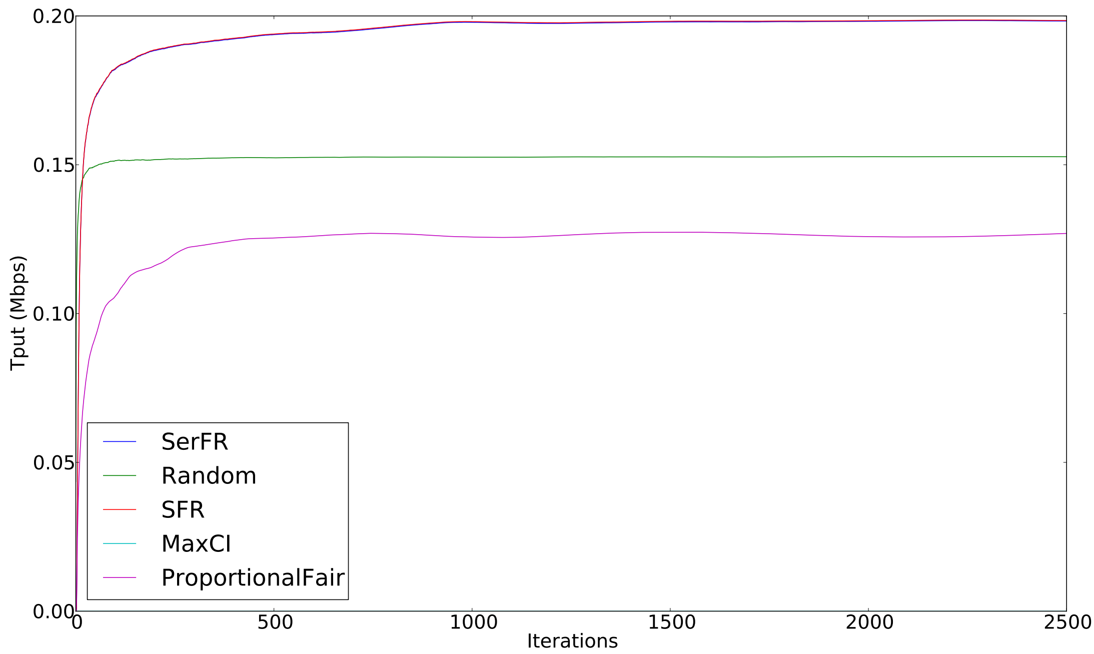


Figure 3.3: The convergence of 5th percentile UE throughput for each algorithm.

### 3.4.2 MO-GA Parameters

The parameters for the MO-GA can be seen in Table 3.8.

The population size and the number of generations were limited as a result of the number of fitness evaluations required to complete an experiment along with

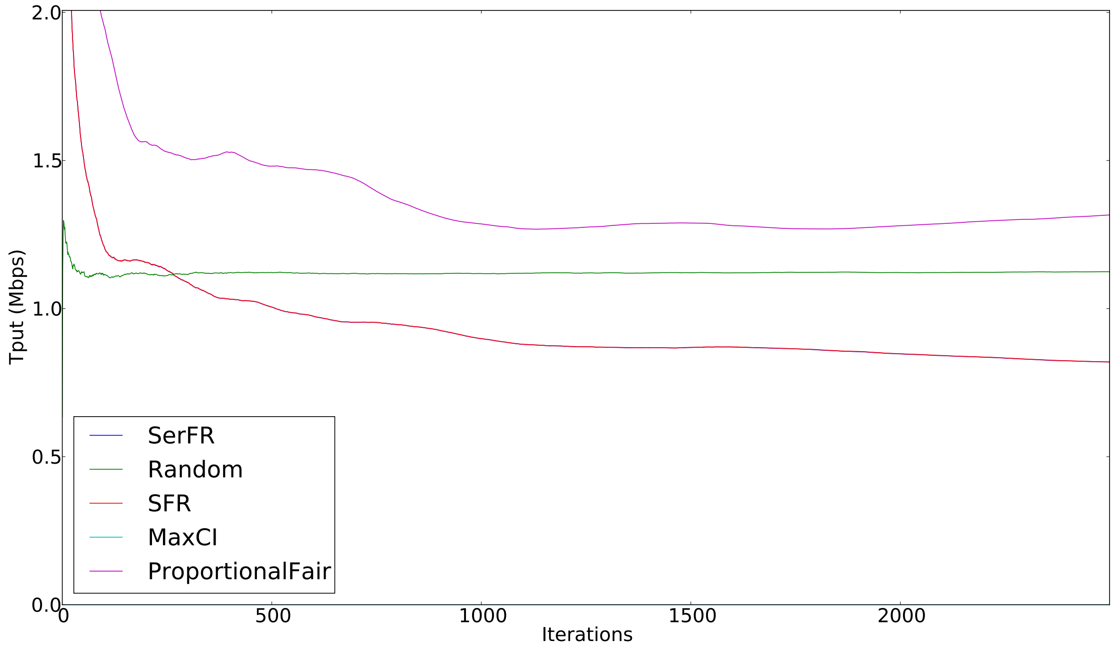


Figure 3.4: The convergence of average UE throughput for each algorithm.

| Parameter | Value |
|---|---|
| Population Size | 10 |
| Generations | 100 |
| Mutation Strength | 1 |
| Mutation Operator | Point Mutation |
| Crossover Operator | Uniform Crossover |

Table 3.8: The parameters used for the MO-GA in experiment 1.

the time taken for each fitness evaluation. With a population of 10 chromosomes and 100 generations a total of 3100 fitness evaluations were required to complete each experiment: 100 evaluations to initialise the population, along with 30 every iteration, 10 for mutation and 20 for the child chromosomes produced by uniform crossover i.e. $100 + (100 \times (10 + 20)) = 3100$

Since the time to execute a single fitness evaluation averaged 3 minutes this results in each MO-GA taking approximately 7 days. Since there were 27 network environments this meant that 27 MO-GAs had to be run. It was decided that this was the upper limit on the amount of time that could be reasonably spent running experiments and so the population and generations were fixed at these values.

The mutation strength indicates how many genes in each chromosome are to be mutated during each iteration of the MO-GA. For further details about the operation of point mutation and uniform crossover refer to Section 2.5.

## 3.5 Results

Due to the fact that a number of algorithms are being compared a mixed ANOVA was run on the results in order to determine the statistical significance of the differences between the algorithms. The between-subjects factor was algorithm and the within-subjects factors were traffic level, speed and number of UEs. This resulted in a 6 (algorithm) $\times$ 4 (traffic level) $\times$ 3 (speed) $\times$ 3 (number of UEs) ANOVA. The average UE throughput and the 5th percentile UE throughput were analysed separately for clarity.

### 3.5.1 Overall Analysis

The overall ANOVA for average UE throughput demonstrated a significant main effect for algorithm, ($F(5,5191) = 1447.98$, $p \leq 0.001$). The F value captures the ratio of variance explained by the factor algorithm to unexplained variance, meaning that a lot of the variance can be attributed to the algorithm. This means that across all conditions the algorithm has a significant effect on the average throughput.

This analysis only shows that different algorithms have different effects on the average throughput. To see which algorithms differed from each other and how significant the differences between each algorithm were a Bonferroni post-hoc analysis was carried out. This analysis performs multiple comparisons between each of the algorithms and adjusts the significance value of any of the differences to account for the fact that there were multiple comparisons. For each comparison the Bonferroni analysis provides a mean difference and the significance of that difference.

As suggested by Figure 3.5 the maximum rate algorithm achieved a significantly higher average UE throughput when compared to all the other algorithms with an average mean difference between the maximum rate algorithm and each of the remaining algorithms of 498kbps (all $p \leq 0.001$). The next best performing algorithms were SerFR, random scheduler, the PF algorithm and the SFR algorithm which were

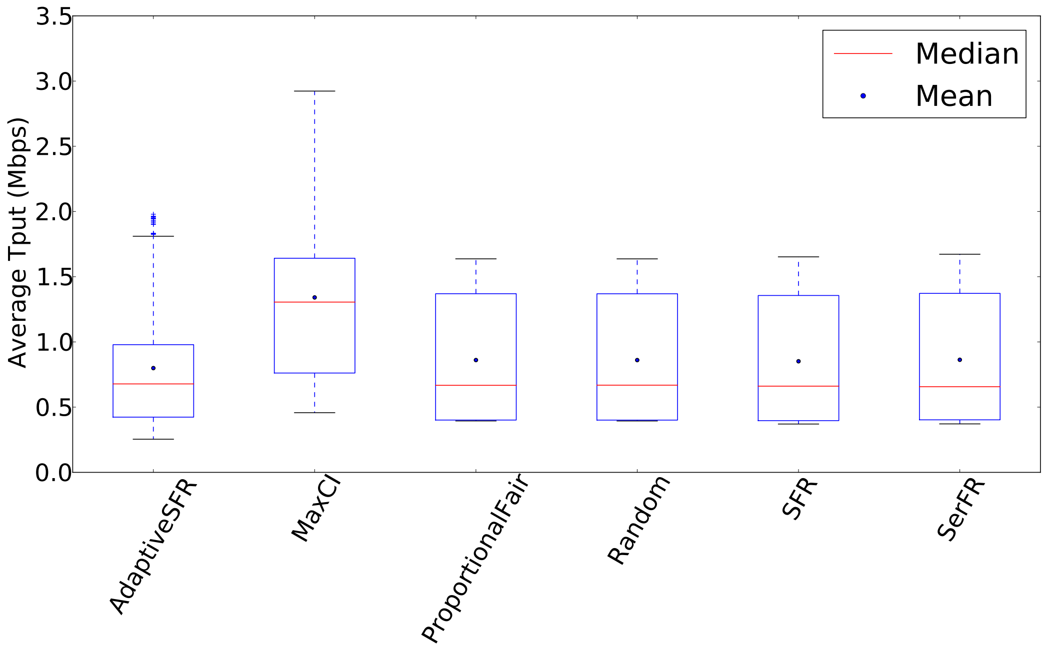


Figure 3.5: Boxplot showing the average UE throughput of each algorithm across all the network environments and traffic levels.

statistically indistinguishable. The aSerFR algorithm performed worst in terms of average UE throughput with an average mean difference between it and the other algorithms, excluding the maximum rate algorithm, of 77kbps (all $p \leq 0.001$).

This overall ANOVA for average UE throughput also demonstrated a number of interaction effects. The interaction effects mean that different algorithms perform differently under different network environments. The strongest interaction was between algorithm and traffic level ($F(15,5191) = 275.21$, $p \leq 0.001$). This called for follow-up simple effects analyses within each level of traffic. These subsequent analyses (ANOVAs investigating the effect of algorithm on average throughput within each traffic level) are detailed in separate sections below.

The same ANOVA was carried out to investigate the effects of algorithm and environment on 5th percentile UE throughput. This demonstrated a significant main effect for algorithm ($F(5,5191) = 1801.82$, $p \leq 0.001$). The Bonferroni post-hoc analysis was then carried out to determine the relative performance of each of the algorithms. The ranking of the algorithms in terms of 5th percentile UE throughput can be seen in Figure 3.6. The performance of each of the algorithms was significantly different from each other, meaning that each algorithm was significantly

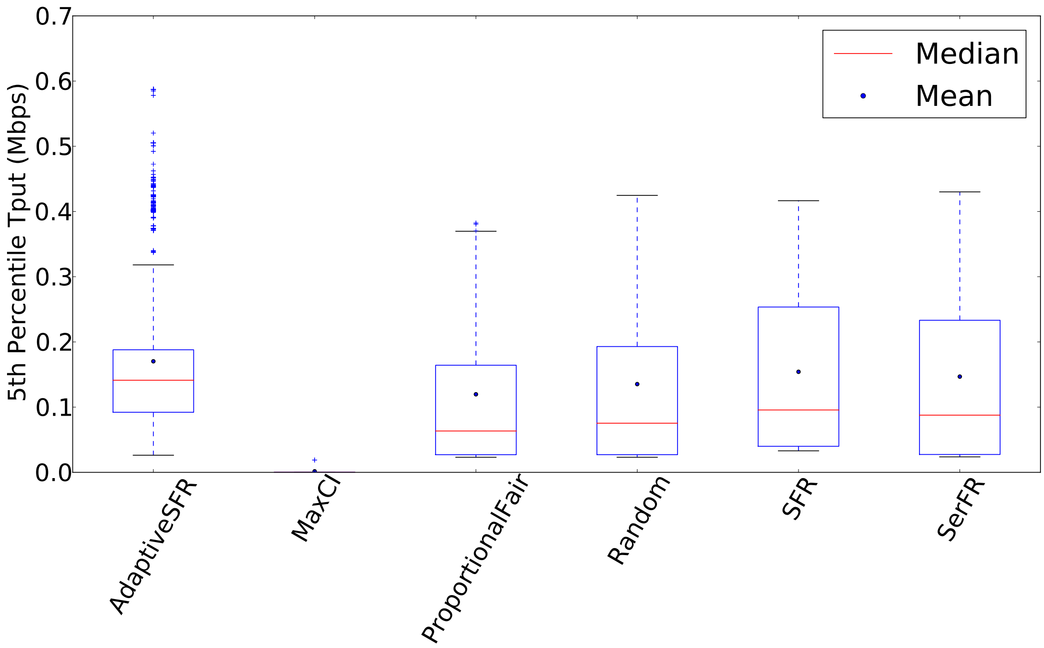


Figure 3.6: Boxplot showing the 5th percentile UE throughput of each algorithm across all the network environments and traffic levels.

better than the next best one. As can be seen the aSerFR outperformed each of the other algorithms, with an average mean difference of 59kbps (all $p \leq 0.001$).

Once again the strongest interaction effect was between algorithm and traffic level calling for separate analyses within each traffic level as detailed below ($F(15,5191) = 243.19$, $p \leq 0.001$).

### 3.5.2 Light Traffic

The results of each algorithm when run under the light traffic network environments are plotted in Figure 3.7. It can be seen from this figure that the average UE throughput of most of the algorithms are very similar, in fact the average UE throughput of the SFR, SerFR, PF and random algorithms are not statistically distinguishable.

A one-way ANOVA looking at the effect of algorithm on average UE throughput within light traffic network environments revealed a significant effect of algorithm ($F(5,1534) = 6.02$, $p \leq 0.001$). As suggested by Figure 3.7 the Bonferroni post-hoc comparisons showed that the only significant difference was between the maximum rate algorithm and the aSerFR algorithm with the maximum rate algorithm per-

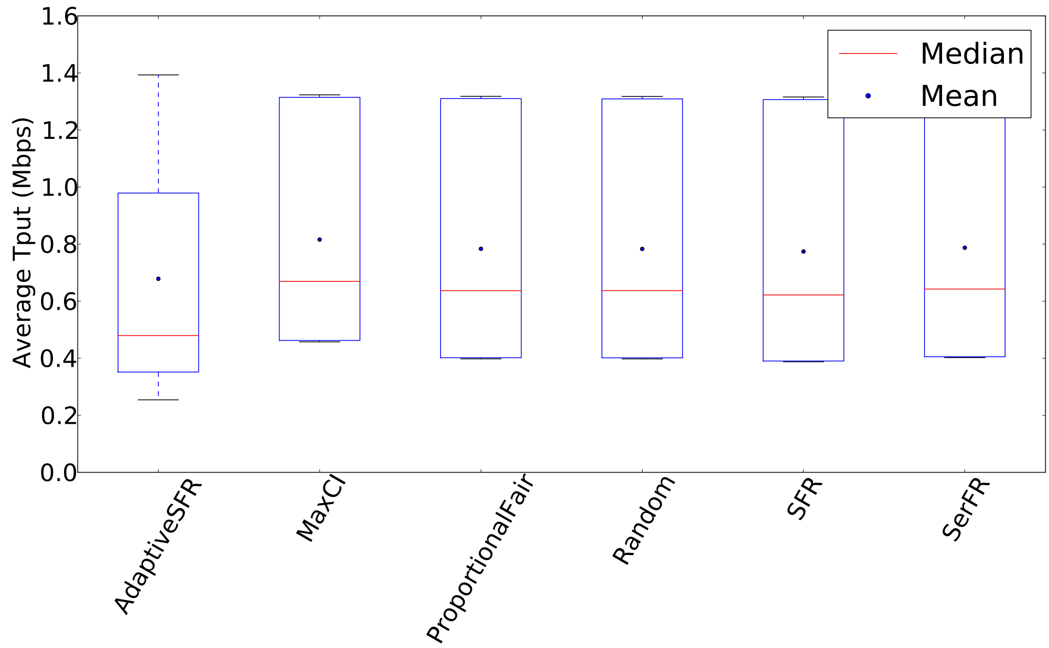


Figure 3.7: Boxplot showing the average throughput of each algorithm across all the network environments under light traffic.

forming significantly better than aSerFR (mean difference = 138kbps, p = 0.01). No other differences in average UE throughput were significant.

Figure 3.8 shows the average performance of each algorithm in terms of the 5th percentile UE throughput. It can be seen that the 5th percentile throughput of the aSerFR algorithm outperformed the other algorithms. Indeed the average 5th percentile throughput was over 100% better than the other algorithms. It can also be seen that the results obtained by the SFR algorithm, while not achieving the performance of the aSerFR algorithm, do outperform those achieved by the other algorithms.

A one-way ANOVA looking at the effect of algorithm on 5th percentile UE throughput within light traffic network environments revealed a significant effect of algorithm ($F(5,1534) = 225.07$, $p \leq 0.001$). As suggested by Figure 3.8 the Bonferroni post-hoc comparisons showed that the aSerFR outperformed each of the other algorithms, with average mean difference of 67kbps (all $p \leq 0.001$). The maximum rate algorithm performed significantly worse than all the other algorithms, with an average mean difference of 36kbps (all $p \leq 0.001$). The remaining algorithms were statistically indistinguishable.

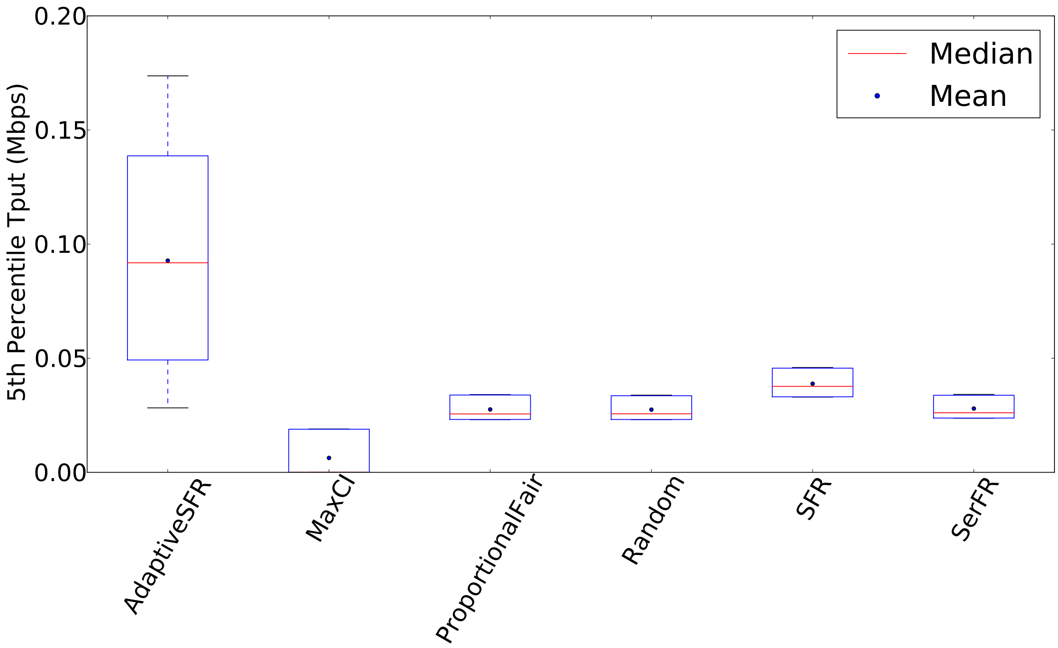


Figure 3.8: Boxplot showing the 5th percentile throughput of each algorithm across all the network environments under light traffic.

The results achieved by each algorithm on the light traffic network environments can be seen in Figure 3.9. This figure shows the nine different network environments run under light traffic. These are combinations of three UE density levels (285 UEs, 680 UEs, 1150 UEs) and three UE speeds (3km/h, 30km/h, 120km/h). These results show that although the aSerFR algorithm performed best on average in terms of 5th percentile UE throughput there are variations in performance depending on the specific network environment. To further investigate the significance of differences between algorithms under specific network environments unpaired t-tests were carried out where appropriate.

It can be seen that in every network environment the aSerFR algorithm outperforms the other algorithms by at least 40kbps (Plot 8) in terms of 5th percentile throughput and in some cases up to 125kbps (Plot 0). In all cases the 5th percentile throughput is improved by at least 100% and in some cases over 300% (Plot 0).

However this increase in 5th percentile throughput comes at a cost to the average UE throughput. The aSerFR chromosomes that result in the best performance in 5th percentile throughput also show a very significant reduction in average throughput, up to 70kbps in some cases.

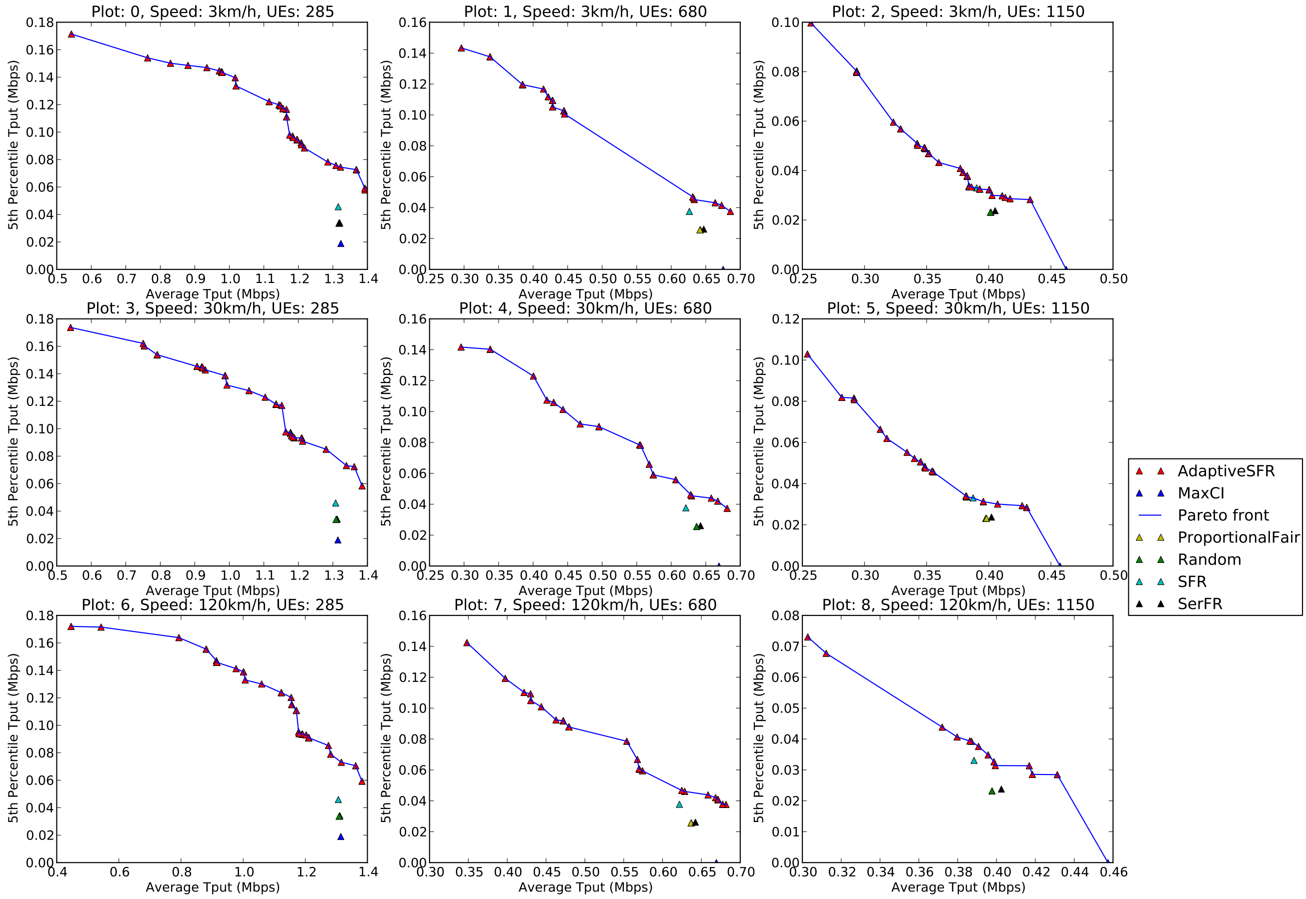


Figure 3.9: The 5th Percentile UE throughput vs average UE throughput in light traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

In some cases the aSerFR algorithm performs better in both 5th percentile throughput and average throughput. This is most clearly seen when there are 285 UEs on the network. For example when there are 285 UEs on the network with speed of 3 km/h (Plot 0) the aSerFR improves on the SFR algorithm (the next best performing algorithm) by 30kbps in 5th percentile throughput with no significant change in average throughput. An independent t-test showed the difference between the aSerFR and SFR algorithms in 5th percentile throughput to be statistically very significant when there are 285 UEs on the network with a speed of 3km/h ($t(18) = 60.3669$, $p = 0.0001$).

It can also be seen from these results that in almost every case the results produced by the aSerFR scheme form a Pareto front which dominates the results produced by all the other schemes. This means that for each result produced by the baseline algorithms there was a result produced by the aSerFR which improved on both the 5th percentile throughput and the average throughput. The exception to this is the maximum rate scheduler which achieves the best average throughput when the network is under the heaviest UE load.

### 3.5.3 Mixed Traffic

As with the light traffic results averages of the results from each algorithm under mixed traffic were taken. The average of the average UE throughput can be seen in Figure 3.10. The average performance of each algorithm under mixed traffic networks displays a similar pattern to the average performance under light traffic loads. In particular the maximum rate algorithm again achieves an improved average UE throughput, with the aSerFR algorithm performing below the other algorithms. The remaining algorithms performed at a very similar level.

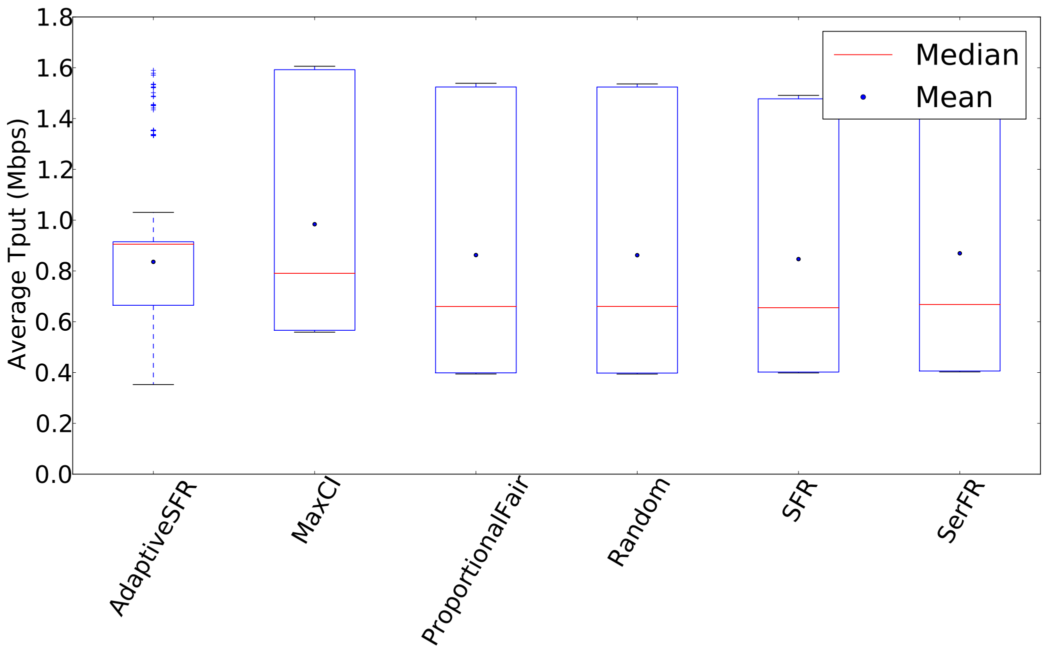


Figure 3.10: Boxplot showing the average throughput of each algorithm across all the network environments under mixed traffic.

As with the light traffic results algorithm was found to have a significant effect on average /glsue throughput (F(5,1406) = 3.07, p = 0.001) by the one-way ANOVA. As suggested in Figure 3.10 the only significant difference found by the Bonferroni post-hoc analysis was between the maximum rate algorithm and the aSerFR algorithm. The maximum rate algorithm performing significantly better than aSerFR (mean difference = 148kbps, p = 0.002). No other differences in average UE throughput were significant.

The results in terms of 5th percentile UE throughput can be seen in Figure 3.11. As with the average UE throughput results these are also very similar to those achieved under light traffic loads. The aSerFR algorithm achieves the best performance by a large margin (over 100% or 40kbps), with the SFR algorithm achieving the next best result. The remaining algorithms, with the exception of the maximum rate achieve a very similar level of performance, while the maximum rate algorithm performs much worse than the others.

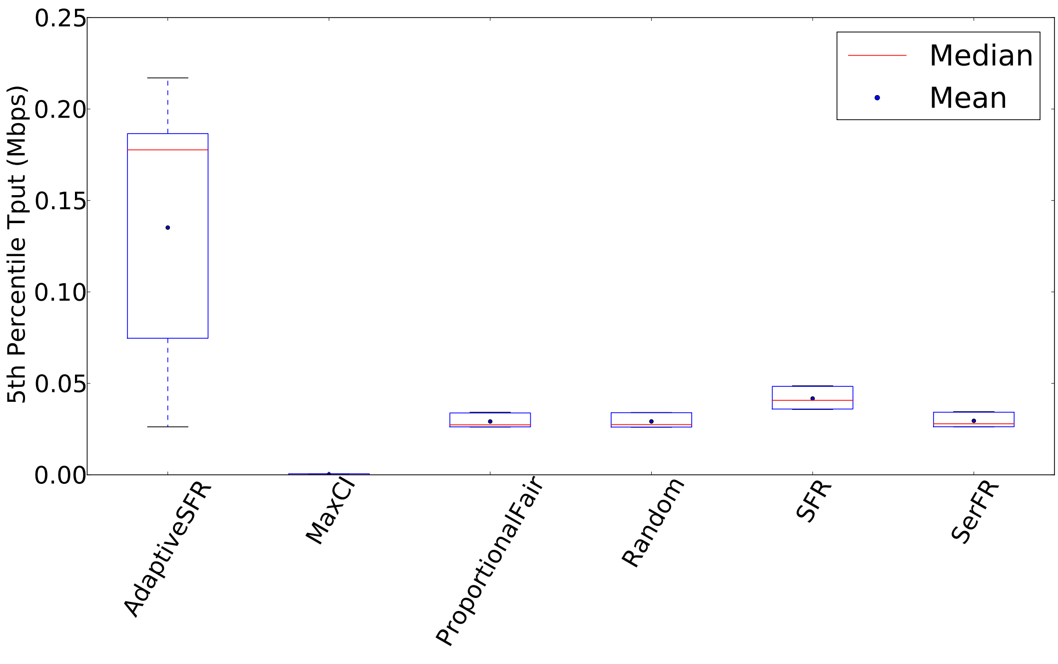


Figure 3.11: Boxplot showing the 5th percentile throughput of each algorithm across all the network environments under mixed traffic.

Algorithm was also found to have a significant effect on the 5th percentile UE throughput ($F(5,1406) = 297.31$, $p \leq 0.001$). with the post-hoc analysis showing that the aSerFR outperformed each of the other algorithms, with average mean difference of 109kbps (all $p \leq 0.001$). The maximum rate algorithm performed significantly worse than all the other algorithms, with an average mean difference of

47kbps (all p $\leq$ 0.001). The remaining algorithms were statistically indistinguishable.

The individual results for each network environment can be seen in Figure 3.12 which shows the results for each algorithm when run under mixed traffic. The results show that, as with light traffic, in each network environment the aSerFR algorithm performed better than the baseline algorithms in terms of 5th percentile throughput with an increase of between 5kbps (Plot 8) up to 169kbps (Plot 0). The improvements are particularly pronounced when either 285 UEs or 680 UEs are on the network. When 1150 UEs are on the network the improvements in 5th percentile throughput are much smaller (approximately 6kbps).

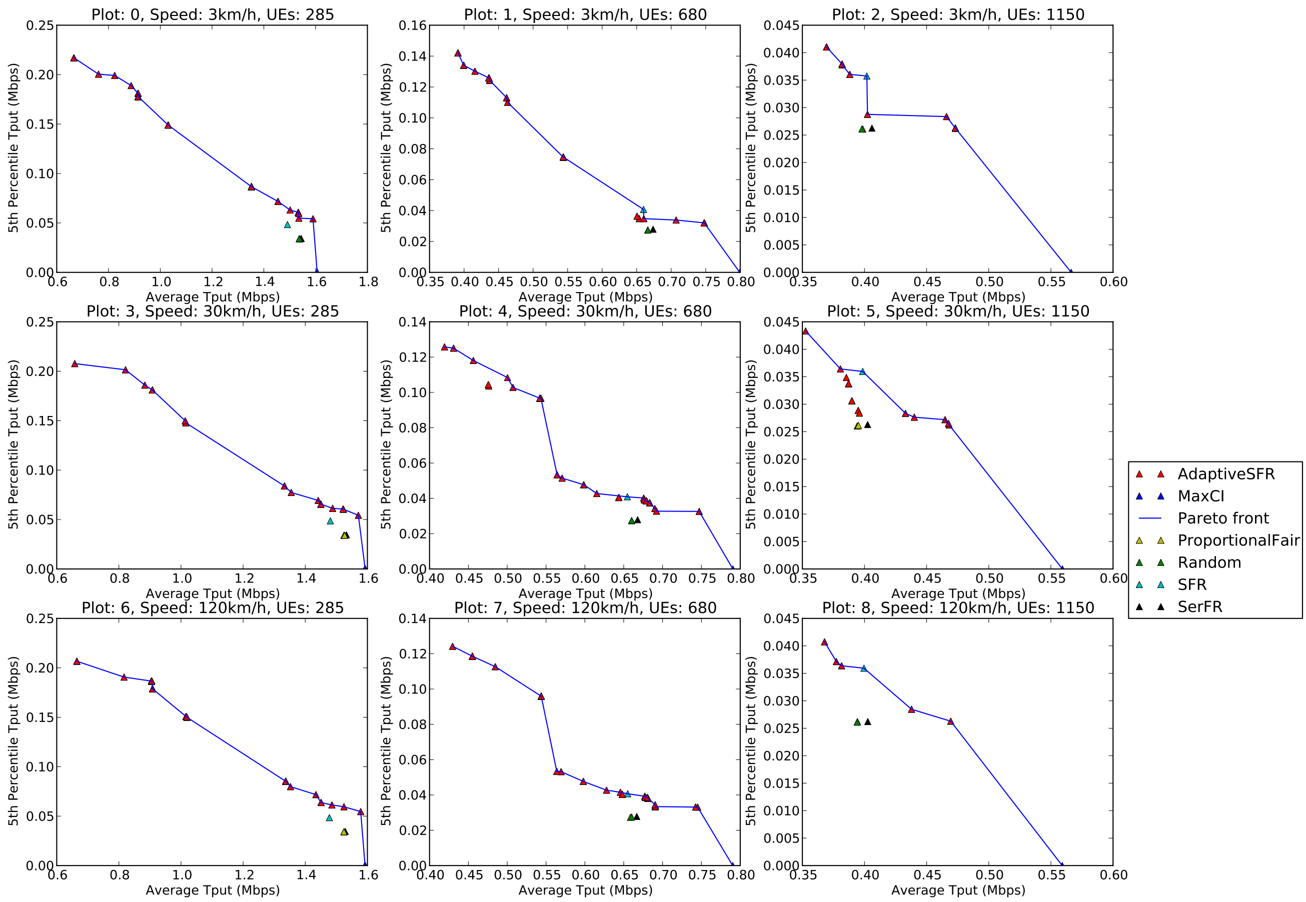


Figure 3.12: The 5th Percentile UE throughput vs average UE throughput in mixed traffic environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

The improvements in 5th percentile throughput are again associated with a reduction in average throughput. The larger the gains in 5th percentile throughput the greater the reduction in average throughput. When the gain in 5th percentile is highest (169kbps in Plot 0) the loss in average throughput is over 800kbps.

It can also be seen that in most cases the results obtained from the aSerFR scheme produced a Pareto front which dominated the solutions produced by the other schemes. The exceptions to this are the maximum rate scheduler which achieves the best average throughput and the SFR scheme contributes to the Pareto front in particular as the network comes under heavier UE load.

### 3.5.4 Heavy Traffic

The graph of the average throughput for each algorithm under Heavy traffic can be seen in Figure 3.13. It can be seen that the maximum rate algorithm achieves the highest performance in terms of the average UE throughput by over 500kbps.

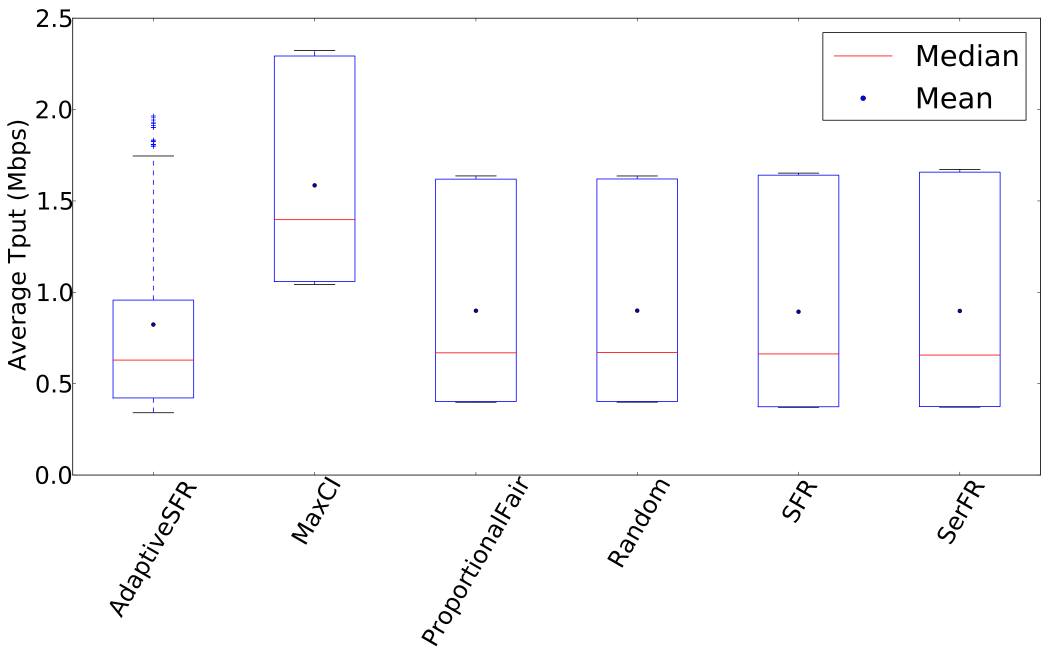


Figure 3.13: Boxplot showing the average throughput of each algorithm across all the network environments under heavy traffic.

Algorithm was again found to be a significant main effect ($F(5,1190) = 34.86$, $p \leq 0.001$). As suggested by Figure 3.13 the Bonferroni post-hoc comparisons showed that the maximum rate algorithm performed significantly better than each of the other algorithms (average mean difference = 702kbps, all p $\leq$ 0.001). All the remaining algorithms are statistically indistinguishable in terms of their average UE throughput.

Figure 3.14 shows the results of each algorithm in terms of the 5th percentile UE throughput. In this case both the SFR and SerFR algorithms performed best. They improved on the results achieved by the other algorithms by at least 25kbps. The aSerFR and random algorithms both achieved a similar level of performance in terms of their 5th percentile UE throughput. The performance of the PF algorithm was about 30kbps less than that of the aSerFR and random algorithms. Finally the maximum rate algorithm achieved an average of 0kbps in terms of 5th percentile throughput.

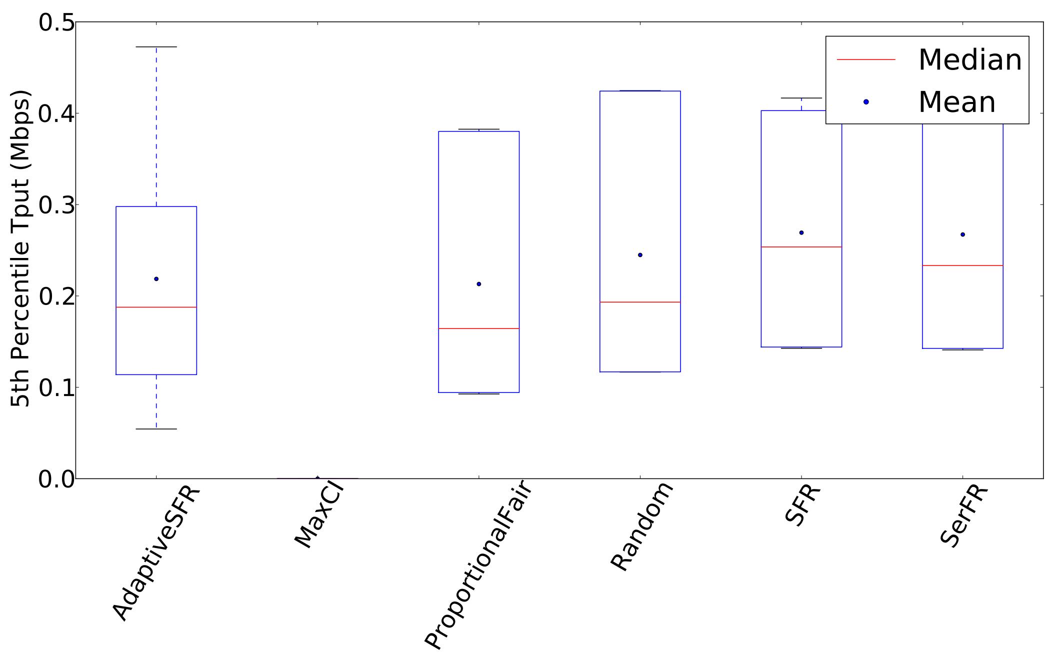


Figure 3.14: Boxplot showing the 5th percentile throughput of each algorithm across all the network environments under heavy traffic.

The one-way ANOVA again confirmed that algorithm had a significant effect on the results ($F(5,1196) = 71.96$, $p \leq 0.001$). As suggested by Figure 3.14 the Bonferroni post-hoc comparisons showed that the SFR and SerFR algorithms performed significantly better than all the other algorithms, but themselves were statistically indistinguishable. The aSerFR algorithm performed at a similar level to the PF and random algorithms, significantly better than the maximum rate algorithm (mean difference of 219kbps, p $\leq$ 0.001) but significantly worse than SFR and SerFR (mean difference of 50kbps, both p $\leq$ 0.002).

As under the previous traffic conditions the results for each algorithm are shown for each network environment in Figure 3.15. As can be seen the results are less clear cut than previously.

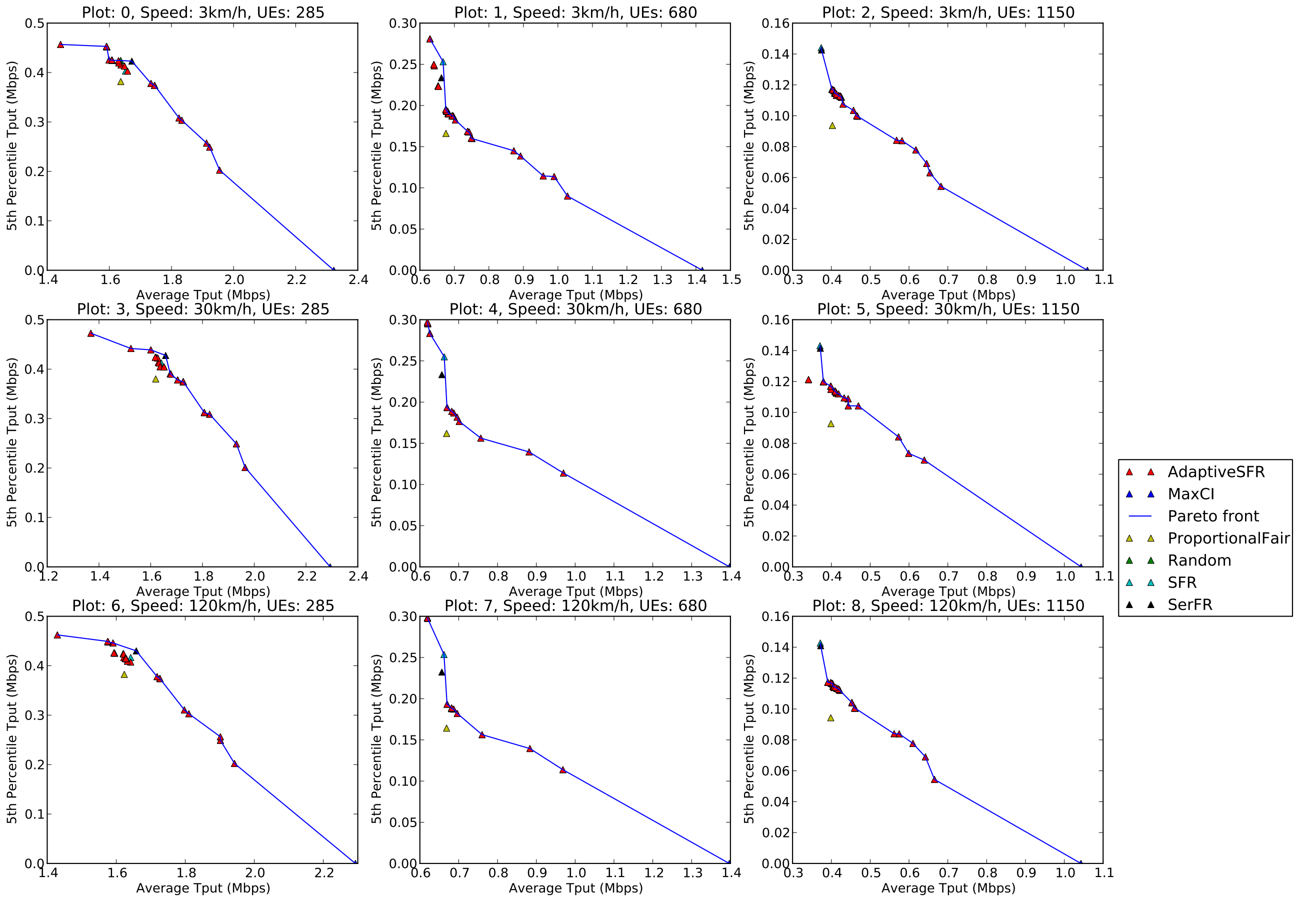


Figure 3.15: The 5th Percentile UE throughput vs average UE throughput in heavy traffic environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

When there are 285 UEs on the network the aSerFR algorithm improves the 5th percentile UE throughput by 25kbps but at a reduction in average UE throughput of 20kbps.

When there are 680 UEs on the network the improvements in 5th percentile throughput achieved by the aSerFR were much smaller, about 5kbps, but still significant. For example when the speed is 120km/h (Plot 7) the aSerFR algorithm performs significantly better than the SFR algorithm ($t(18) = 3.1817$, $p = 0.0052$). The decrease in average UE throughput of the aSerFR by 30kbps compared to the SFR in this case is also statistically significant ($t(18) = 2.9616$, $p = 0.0084$).

With 1150 UEs on the network the aSerFR no longer shows any improvement in terms of either 5th percentile or average throughput. For example in Plot 8 the SFR algorithm performs better than the aSerFR algorithm by a significant 27kbps ($t(18) = 7.4401$, $p = 0.0001$).

In each case the Pareto front receives contributions from most of the algorithms suggesting that no one approach is able to provide overall optimal performance. In every case the PF algorithm is dominated by another solution on the Pareto front showing that in every case there is at least one solution which achieves both better 5th percentile throughput and better average throughput than the PF algorithm.

### 3.5.5 Full Traffic

The average UE throughput results obtained under Heavy traffic can be seen in Figure 3.16. It can be seen that the maximum rate algorithm outperforms the other algorithms by a very large margin, over 100% or 1Mbps. The other algorithms achieve a very similar level of performance.

Algorithm was again found to be the main effect ($F(5,1253) = 69.35$, $p \leq 0.001$). As suggested by Figure 3.16 the Bonferroni post-hoc comparisons showed that the maximum rate algorithm performed significantly better than each of the other algorithms (average mean difference = 1.083Mbps, all $p \leq 0.001$). All the remaining algorithms are statistically indistinguishable in terms of their average UE throughput.

The 5th percentile throughput results, seen in Figure 3.17 show that the best performing algorithms, aSerFR, SFR and SerFR achieve a very similar level of performance with the random and PF algorithms achieving worse results by 20kbps and 50kbps respectively.

As before algorithm was found to be the main effect($F(5,1253) = 72.82$, $p \leq$

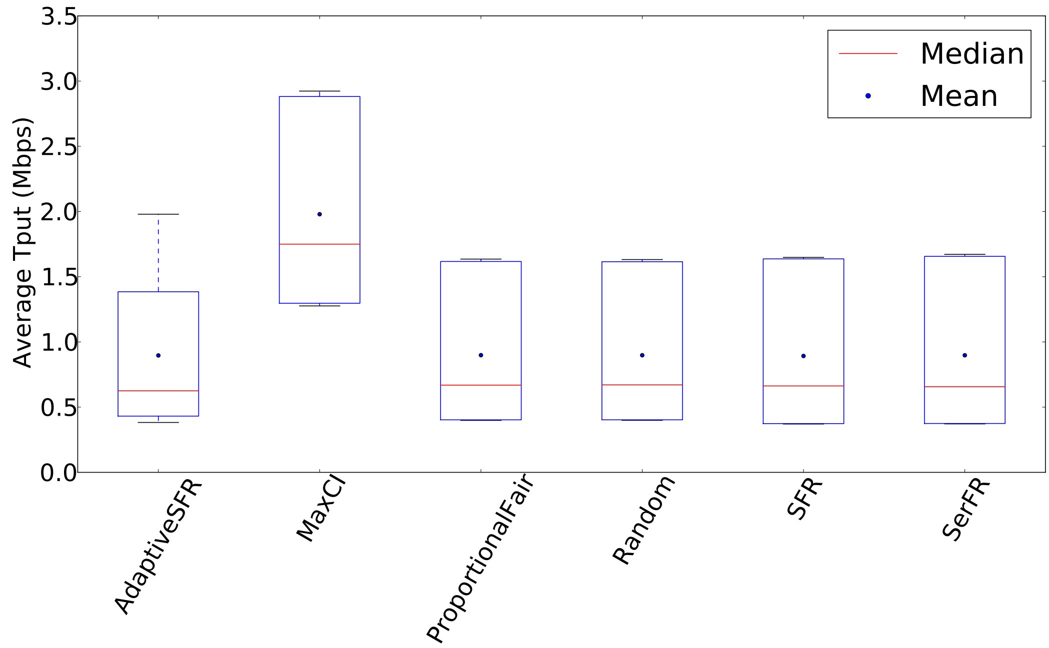


Figure 3.16: Boxplot showing the average throughput of each algorithm across all the network environments under full traffic.

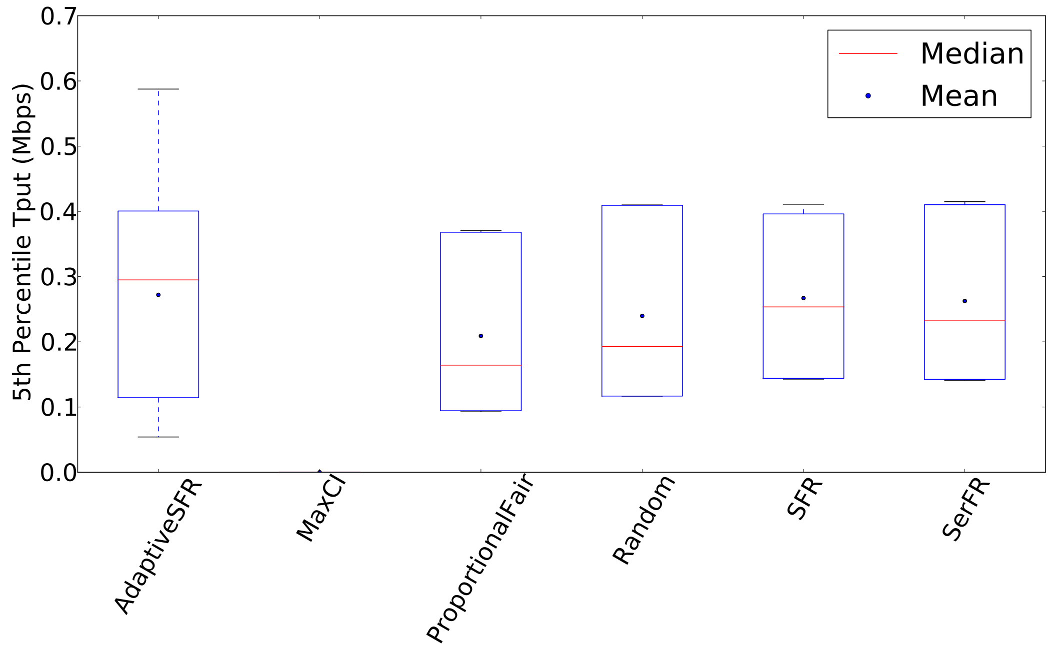


Figure 3.17: Boxplot showing the 5th percentile throughput of each algorithm across all the network environments under full traffic.

0.001). As suggested by Figure 3.17 the aSerFR provided the best performance. The Bonferroni post-hoc comparisons showed that the aSerFR was only significantly better than the maximum rate algorithm (mean difference of 272kbps, $p \leq 0.001$) and the PF (mean difference of 63kbps, $p \leq 0.001$). As before the maximum rate performs significantly worse than all the other algorithms with an average mean difference of 250kbps (all p $\leq$ 0.001).

As before Figure 3.18 shows the results achieved by each algorithm under each network environment.

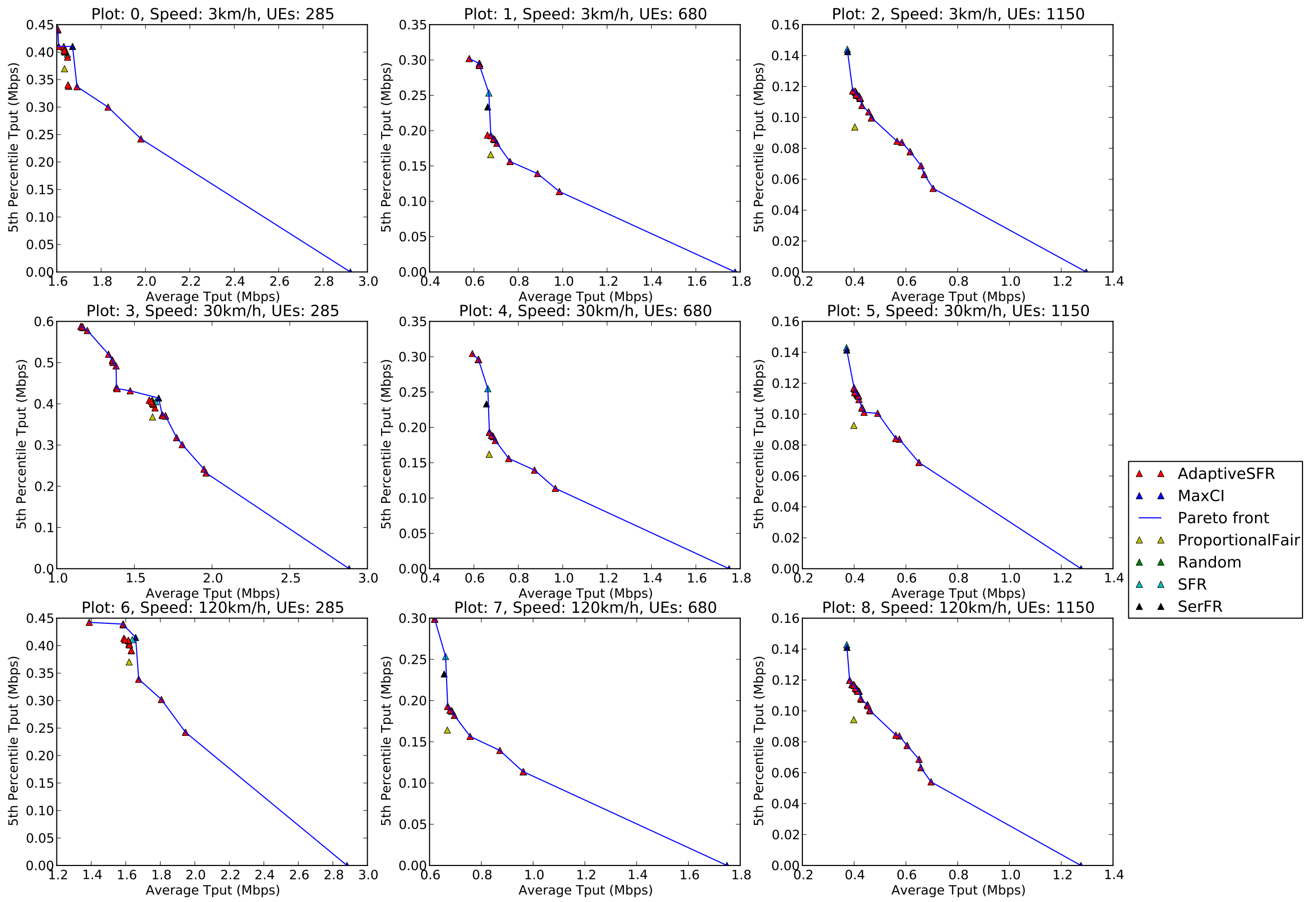


Figure 3.18: The 5th Percentile UE throughput vs average UE throughput in full traffic environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

When 285 UEs are present on the network the aSerFR performs significantly better in terms of 5th percentile throughput than the baseline algorithms. When the speed is either 3km/h (Plot 0) the performance of the aSerFR is significantly better than that of the SerFR algorithm in terms of 5th percentile throughput ($t(18) = 2.4178$, $p \leq 0.0264$) with no statistically significant reduction in average UE throughput ($t(18) = 1.0725$, $p = 0.2977$). However when the speed is 30km/h

(Plot 3) the performance of the aSerFR is significantly greater, by 173kbps, than the performance achieved by SerFR ($t(18) = 13.0365$, $p = 0.0001$), however this comes with a significant reduction of 400kbps in average UE throughput between SerFR and aSerFR ($t(18) = 10.5249$, $p = 0.0001$).

When 680 UEs are present on the network the aSerFR performs significantly better in terms of 5th percentile throughput by providing an improvement of 50kbps over the baseline algorithms. For example in (Plot 7) the aSerFR algorithm performed significantly better than the SFR algorithm ($t(18) = 6.0910$, $p = 0.0001$). This comes with a statistically significant reduction in average throughput from 665kbps with SerFR to 620kbps with aSerFR ($t(18) = 3.3006$, $p = 0.004$).

With 1150 UEs on the network both the SFR and SerFR algorithms perform best. For example in Plot 8 the SFR algorithm improves on the performance achieved by the aSerFR algorithm by a significant 23kbps ($t(18) = 5.8800$, $p = 0.0001$).

The Pareto front produced in this case shows that when the traffic is full the SerFR algorithm dominates the results produced by a number of the other algorithms including, in some cases, the aSerFR algorithm. As with the mixed traffic scheme the results produced by the PF algorithm are always dominated by those produced by another algorithm under the full traffic load.

### 3.5.6 Summary

The results described in the preceding sections demonstrate that the aSerFR algorithm was in general able to improve significantly on the performance of the 5th percentile UE throughput in most cases. Further analysis showed that the improvements made under light and mixed traffic levels were both significant and substantial. Under heavy traffic the aSerFR algorithm matched the performance of the best existing baseline algorithms in terms of the 5th percentile UE throughput, but under heavy traffic the aSerFR was not able to achieve the same performance as the baseline algorithms.

In terms of the average UE throughput the aSerFR performed significantly worse overall. In general the lighter the traffic level the worse the relative performance of the aSerFR was. Under full traffic environments the aSerFR was indistinguishable from the other baseline algorithms, except for the maximum rate algorithm which consistently performed best in terms of average UE throughput.

## 3.6 Discussion

This chapter introduced and described a new approach to ICIC in LTE networks. This approach, known as aSerFR, was based on the existing scheme SerFR and allowed a GA to optimise its parameters in order to tailor its behaviour to various network environments. This was discussed as an example of localised organisation where the eNBs do not have any knowledge of their neighbours' behaviour and there is no active coordination. An experiment was described which compared this approach to existing ICIC schemes and some baseline schedulers in order to see if it offered any improvement in the performance of the network.

The results described in the previous section demonstrate that particular improvements were noted when the network was under one of the lower traffic loads, either light or mixed. In these cases it was observed that aSerFR achieved significantly better 5th percentile throughput. In many cases aSerFR dominated all the other algorithms, achieving both better 5th percentile throughput along with better average throughput. Under these lighter traffic loads there is less contention on the network's resources and under these conditions aSerFR utilises the available resources more effectively than the baseline algorithms.

The improvements achieved by aSerFR under lighter traffic loads can also be seen to a lesser extent when the network is under heavier traffic loads. When there are less UEs on the network the aSerFR does achieve some improvement in 5th percentile throughput, however the improvements are more modest than under lighter traffic loads. However when there are more UEs active on the network the SFR and SerFR schemes dominate by a significant margin in every case.

The results imply that localised organisation demonstrated in the aSerFR scheme can indeed offer improvements in the performance of the network. The improvements are greatest when the radio resources of the network are relatively uncontested showing that this type of organisation can achieve good results when the constraints are looser. When the constraints on the network resources are more severe however it can be seen that this type of organisation does not improve the 5th percentile throughput but rather the existing SFR and SerFR schemes are best suited to this environment.

In summary it can be seen that this type of local organisation can lead to significant improvements in 5th percentile UE throughput, particularly under light and mixed traffic loads. Under heavier traffic loads this approach was shown to match the performance of the existing ICIC approaches.

# Chapter 4

# Networked Organisation for eNodeB Co-ordination

In Chapter 3 a new Inter-Cell Interference Coordination (ICIC) scheme was introduced which incorporated the idea of localised organisation to reduce the interference in Long Term Evolution (LTE) networks. This was achieved by using ac Genetic Algorithm (GA) to optimise it for use on a variety of network environments. It was observed that this type of local organisation offered some improvements in network performance when compared to existing ICIC schemes such as Soft Frequency Reuse (SFR) and Softer Frequency Reuse (SerFR), in particular under light and mixed traffic loads. In the local organisation demonstrated in the previous chapter the network nodes, Evolved NodeBs (eNBs), did not have any knowledge of their neighbours' behaviour, but rather optimised their own behaviour.

One of the key features of LTE networks in the context of this research is the X2 link which can be used by eNBs to communicate directly with their neighbours. This chapter explores the possibilities of utilising this X2 link to enable a more networked type of organisation. It is expected that allowing the eNBs to communicate with their immediate neighbours to share scheduling information will lead to better network performance as the eNBs will be able to take into account the performance of their neighbours when making scheduling decisions.

This chapter describes the X2 protocol which operates in LTE networks and a new ICIC scheme which is built on the Adaptive Softer Frequency Reuse (aSerFR) scheme presented in the previous chapter and utilises this X2 link to network the eNBs so that they can schedule their transmissions in a way that attempts to reduce the interference experienced by the network.

## 4.1 Overview of the X2 Protocol

The X2 protocol is a point-to-point protocol which enables eNBs to directly communicate with each other. This direct communication allows the eNBs to exhibit a greater level of autonomy than was available under Universal Mobile Telecommunications System (UMTS) by sharing information which could allow for self-organisation and self-optimisation. The main use cases envisaged by the 3rd Generation Partnership Project (3GPP) are for mobility support, load management, ICIC and error reporting [7, 23].

Mobility management is concerned with the movement of User Equipments (UEs) from one cell to another. By communicating with each other it is expected that the eNBs can manage handovers in a more seamless manner.

The load management features allow the eNBs to share information about the interference levels experienced on their Resource Blocks (RBs).

The details of the X2 protocol are defined in a number of 3GPP standards documents [11, 12, 13, 18, 14]. It is important to note that the standards that define these protocols are still under active development and that while 3GPP is responsible for establishing the protocol standards, in this case what messages will be passed along the X2 protocol, they do not enforce the behaviour an eNB should take in each instance. These implementation details are left to the vendor to determine. There is also a possibility of developing additions to the X2 functionality which may or may not be incorporated into future standards. For example an eNB vendor may develop a proprietary message which will run over the X2 link and which will be usable only by their eNBs.

## 4.2 Adaptive Softer Frequency Reuse with X2

The ICIC scheme presented in this section, known as Adaptive Softer Frequency Reuse + X2 (aSerFR+X2) utilises the X2 link to share information about the downlink interference. aSerFR+X2 defines a new type of message which can be used to share information regarding downlink interference between eNBs. As with the aSerFR scheme a number of parameters have been specified so that this scheme can be optimised in a similar way. It functions in the following way.

When an eNB becomes heavily loaded it can send a message to its neighbours containing a list of those RBs which are experiencing the greatest level of interference. The number of RBs included in such a message is dictated by the *numRBs* parameter. Upon receiving such a message an eNB will avoid scheduling any transmissions on those RBs. Such a message can be sent by an eNB when its load level rises above a certain level which is determined by the *loadLevel* parameter. The message will either be sent to a randomly chosen neighbour or all neighbours depending on the value of the *neighbours* parameter. Each X2 message sent by an eNB will have a certain lifetime during which it will be in effect. This is governed by the $msgLifetime$ parameter. To prevent a heavily loaded eNB from continuously sending out messages to its neighbours the $msgWaitTime$ parameter defines the length of time an eNB must wait before sending another X2 message.

## 4.3 Experiments

In order to evaluate the performance of the aSerFR+X2 scheme presented in this chapter and to compare it against existing approaches a number of experiments were run. As in Chapter 3 the ICIC schemes were compared across 27 different network environments in order to build up a more complete picture of their performance characteristics. The network environments varied by the number of UEs present on the network, the speed of the UEs on the network and the level of traffic requested by the UEs.

As in experiment 1 described in Section 3.4 an Multi-Objective Genetic Algorithm (MO-GA) was run for each algorithm in order to optimise the parameters of the aSerFR+X2 scheme and achieve the best performance. The MO-GA produced a number of results for each network environment which, taken with the results of the other algorithms, produce a Pareto front where all the solutions can be considered equally fit, but with varying values of 5th percentile and average throughput.

The MO-GA was run for 100 generations and the simulator described in Section 3.2 was used to evaluate the fitness of each of the chromosomes in the population. In order to ensure that the fitness value calculated for any given chromosome was representative of that chromosome's fitness the simulator was run ten times with different random seeds and the results averaged to obtain the fitness. The remaining parameters of the MO-GA are shown in Table 4.1. The parameters used in the simulator are identical to those used in Chapter 3. See Table 4.2 for reference.

| Parameter | Value |
|---|---|
| Population Size | 10 |
| Generations | 100 |
| Mutation Strength | 1 |
| Mutation Operator | Point Mutation |
| Crossover Operator | Uniform Crossover |

Table 4.1: The parameters used for the MO-GA in experiment 2.

| Parameter | Value |
|---|---|
| Number of eNBs | 19 |
| Number of UEs | 252, 680, 1150 |
| UE speed | 3kmh, 30kmh, 120kmh |
| Data generation | Light, Mixed, Heavy Traffic profiles |
| Distance attenuation | $L = 128.1 + 37.6 \times log(distance)$ [6] |
| Shadow fading | Log normal, 8dB standard deviation |
| Multi-path fading | Rayleigh fading |
| Cell layout | Hexagonal grid, 3-sector sites, 57 sectors in total |
| Cell Radius | 500m inter-site distance |
| Simulation Time | 500 ms |

Table 4.2: The parameters used for the simulator in experiment 2.

## 4.4 Results

As with the results of the previous experiment this section will report on the performance of the aSerFR+X2 algorithm in comparison to those algorithms already presented.

As in the previous chapter due to the fact that a number of algorithms are being compared an mixed ANOVA was run on the results in order to determine the statistical significance of the differences between the algorithms. The between-subjects factor was algorithm and the within-subjects factors were traffic level, speed and number of UEs. This resulted in a 6 (algorithm) $\times$ 4 (traffic level) $\times$ 3 (speed) $\times$ 3 (number of UEs) ANOVA. The average UE throughput and the 5th percentile UE throughput were analysed separately for clarity.

### 4.4.1 Overall Analysis

The overall ANOVA for average UE throughput demonstrated a significant main effect for algorithm, ($F(6,6620) = 1292.11$, $p \leq 0.001$). This shows that much of the variance in the results can be attributed to the algorithm which means that the algorithm has a significant effect on the average UE throughput.

As before a Bonferroni post-hoc analysis was carried out to determine which of the algorithms differed from each other and how significant the differences between each algorithm were. This analysis performs multiple comparisons between each of the algorithms and adjusts the significance value of any of the differences to account for the fact that multiple comparisons have taken place. For each comparison the Bonferroni analysis provides a mean difference and the significance of that difference.

As suggested by Figure 4.1 the maximum rate algorithm achieved a significantly higher average UE throughput when compared to all the other algorithms with an average mean difference between the maximum rate algorithm and each of the remaining algorithms of 518kbps (all $p \leq 0.001$). The next best performing algorithms were SerFR, random scheduler, the Proportional Fair (PF) algorithm and the SFR algorithm which were statistically indistinguishable. The aSerFR algorithm performed worse again that the above algorithms in terms of average UE throughput with an average mean difference between it an the other algorithms, excluding the maximum rate algorithm, of 124kbps (all $p \leq 0.001$). Finally the aSerFR+X2 algorithm performed significantly worse than the aSerFR algorithm. The mean difference

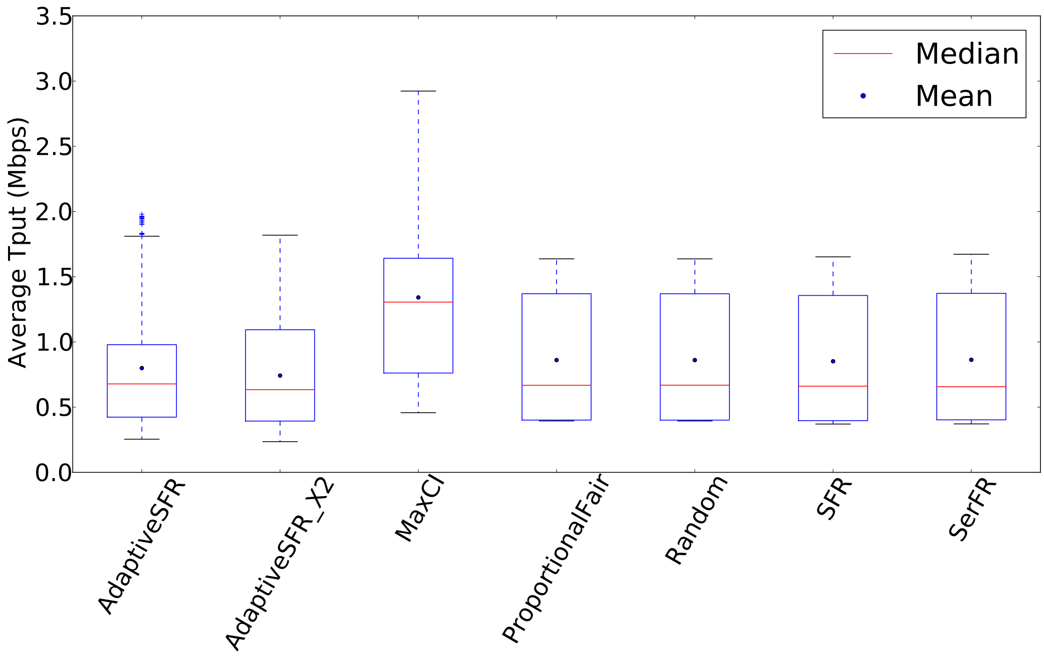


Figure 4.1: Boxplot showing the average UE throughput of each algorithm across all all traffic levels.

was 62kbps (p $\leq$ 0.001).

This overall ANOVA for average UE throughput also demonstrated a number of interaction effects. The interaction effects mean that different algorithms perform differently under different network environments. The strongest interaction was between algorithm and traffic level (F(18,6620) = 207.98, p $\leq$ 0.001). As before this called for follow-up simple effects analyses within each level of traffic. These subsequent analyses (ANOVAs investigating the effect of algorithm on average throughput within each traffic level) are detailed in separate sections below.

The results of the algorithms in terms of 5th percentile UE throughput can be seen in Figure 4.2. The same ANOVA was carried out to investigate the effects of algorithm and environment on 5th percentile UE throughput. This demonstrated a significant main effect for algorithm (F(6,6620) = 5030.61, p $\leq$ 0.001). The Bonferroni analysis showed that most of the algorithms produced significantly different results. In particular the aSerFR+X2 algorithm performed significantly better than all the other algorithms, with an average mean difference of 66kbps (all p $\leq$ 0.001).

Once again separate analyses were conducted within each traffic level in view of the significant interaction between traffic level and algorithm (F(18,6620) = 131.83,

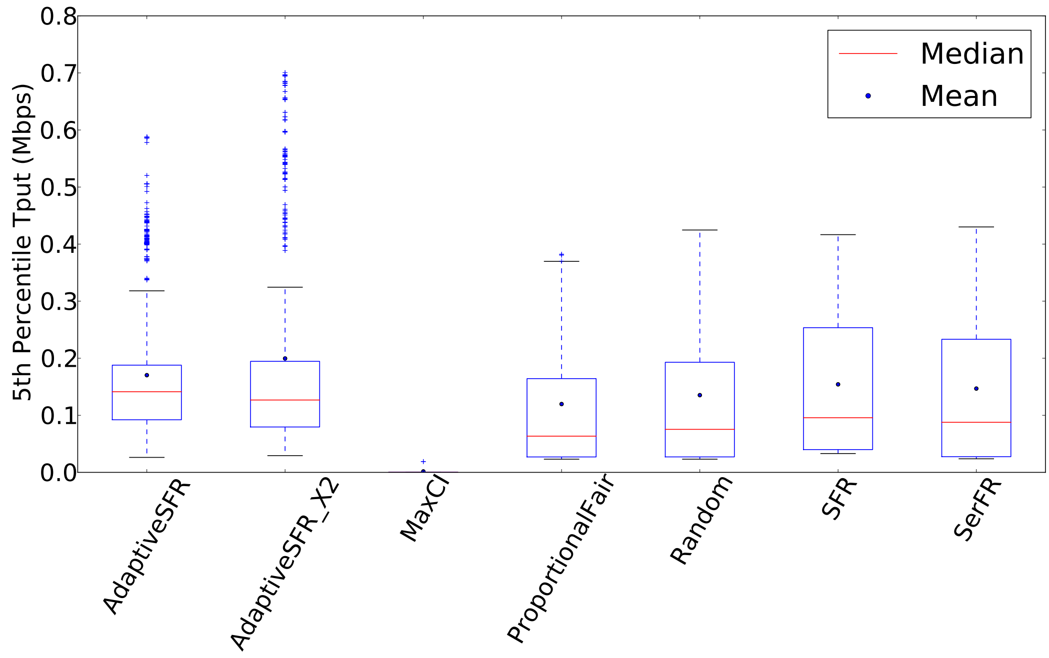


Figure 4.2: Boxplot showing the 5th percentile UE throughput of each algorithm across all all traffic levels.

p $\leq$ 0.001).

### 4.4.2 Light Traffic

The results of each algorithm when run under the light traffic network environments can be seen in Figure 4.3. It can be seen from this figure that, as expected, the relative performance of each of the algorithms is very similar to that described in Chapter 3. The aSerFR+X2 algorithm introduced in this chapter performs worse than all the other algorithms tested.

Algorithm was again found to be a significant effect on the results (F(6,1958) = 20.20, p $\leq$ 0.001). As suggested by Figure 3.7 the Bonferroni post-hoc comparisons showed that the aSerFR+X2 algorithm performed significantly worse than all the other algorithms with an average mean difference of 237kbps (all p $\leq$ 0.001). The aSerFR algorithm performed significantly better than the aSerFR+X2 with a mean difference of 146kbps (p $\leq$ 0.001) and significantly worse than the maximum rate algorithm with a mean difference of 138kbps (p = 0.006). No other differences were significant.

It should be noted here that the level of variance in the results has changed

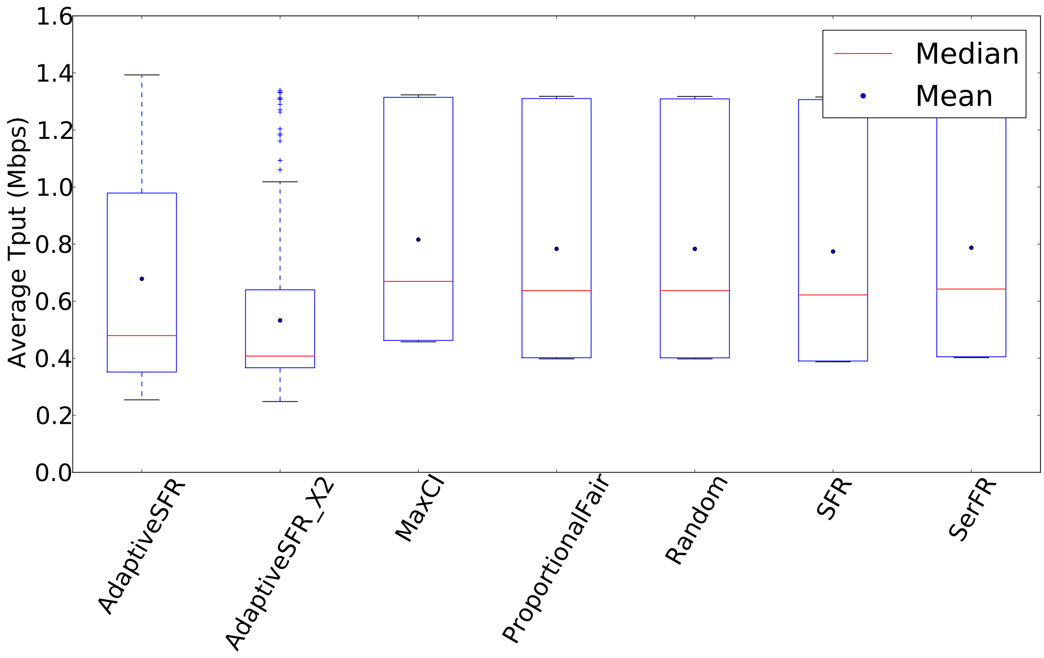


Figure 4.3: Boxplot showing the average UE throughput of each algorithm across all light traffic network environments.

from the previous chapter due to the addition of another algorithm (aSerFR+X2) into the multiple comparisons. In the previous chapter the aSerFR was found to be significantly worse than the baseline algorithms in this case, but with the addition of the aSerFR+X2 algorithm to the comparisons, which performs worst of all, the difference between the aSerFR and the baseline algorithms was found to be insignificant.

Figure 4.4 shows the average performance of each algorithm in terms of the 5th percentile UE throughput. It can be seen that the 5th percentile throughput of the aSerFR+X2 algorithm outperformed the other algorithms. It can also be seen that the results obtained by the SFR algorithm, while not achieving the performance of the aSerFR+X2 or aSerFR algorithm, do outperform those achieved by the other algorithms.

As with the average throughput results algorithm was again found to be a significant effect ($F(6,1958) = 239.06$, $p \leq 0.001$). As suggested by Figure 4.4 the Bonferroni post-hoc comparisons showed that the aSerFR+X2 outperformed each of the other algorithms, with average mean difference of 69kbps (all $p \leq 0.001$).

As in the previous chapter the results achieved by each algorithm on the light

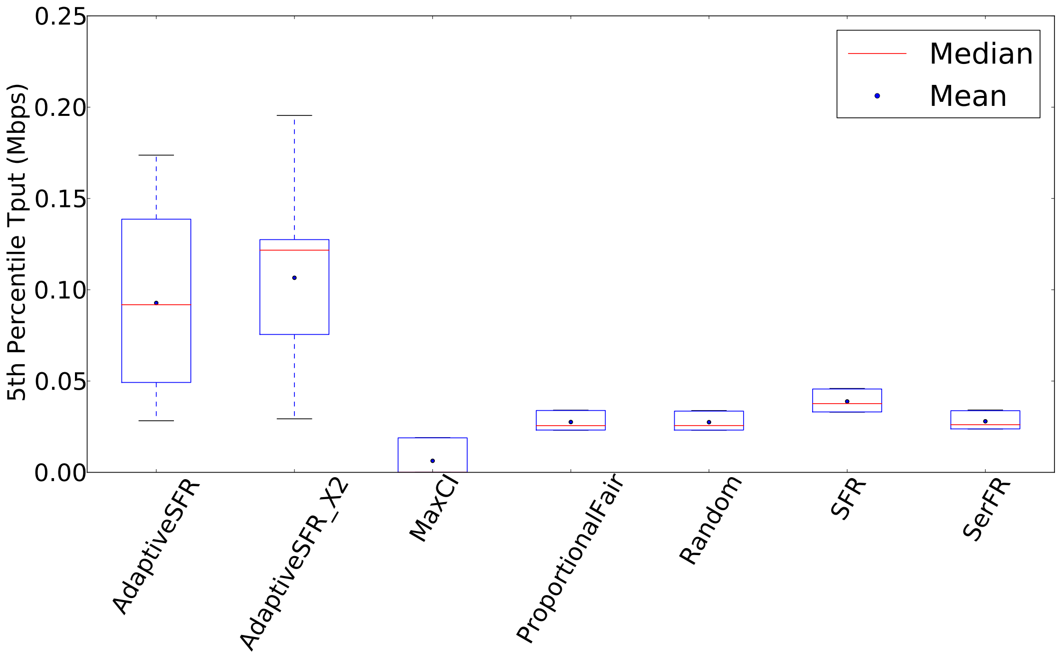


Figure 4.4: Boxplot showing the average 5th percentile UE throughput of each algorithm across all light traffic network environments.

traffic network environments can be seen in Figure 4.5. This figure shows the nine different network environments run under light traffic. These are combinations of three UE density levels (285 UEs, 680 UEs, 1150 UEs) and three UE speeds (3km/h, 30km/h, 120km/h). It can be seen that when the network is under a light traffic load and there are relatively few UEs the aSerFR+X2 scheme generally improves on the performance achieved by the aSerFR and other algorithms. For example in Plot 6 the aSerFR+X2 algorithm outperforms the aSerFR algorithm by a statistically significant 23kbps ($t(18) = 6.3196$, $p = 0.0001$).

With 680 users the 5th percentile throughput values achieved by the aSerFR+X2 do not surpass those achieved by the aSerFR. However in several cases it can achieve a similar 5th percentile throughput with a greater average UE throughput. For instance when there are 680 UEs on the network with speed 120km/h (Plot 7) the aSerFR can achieve a 5th percentile throughput of 104kbps with an average throughput of 430kbps. In comparison the aSerFR+X2 can achieve the same 5th percentile throughput while improving the average throughput by 70kbps.

When the user load is increased to 1150 UEs the aSerFR+X2 achieves the best absolute 5th percentile throughput only when the UE speed is 120km/h (Plot 8).

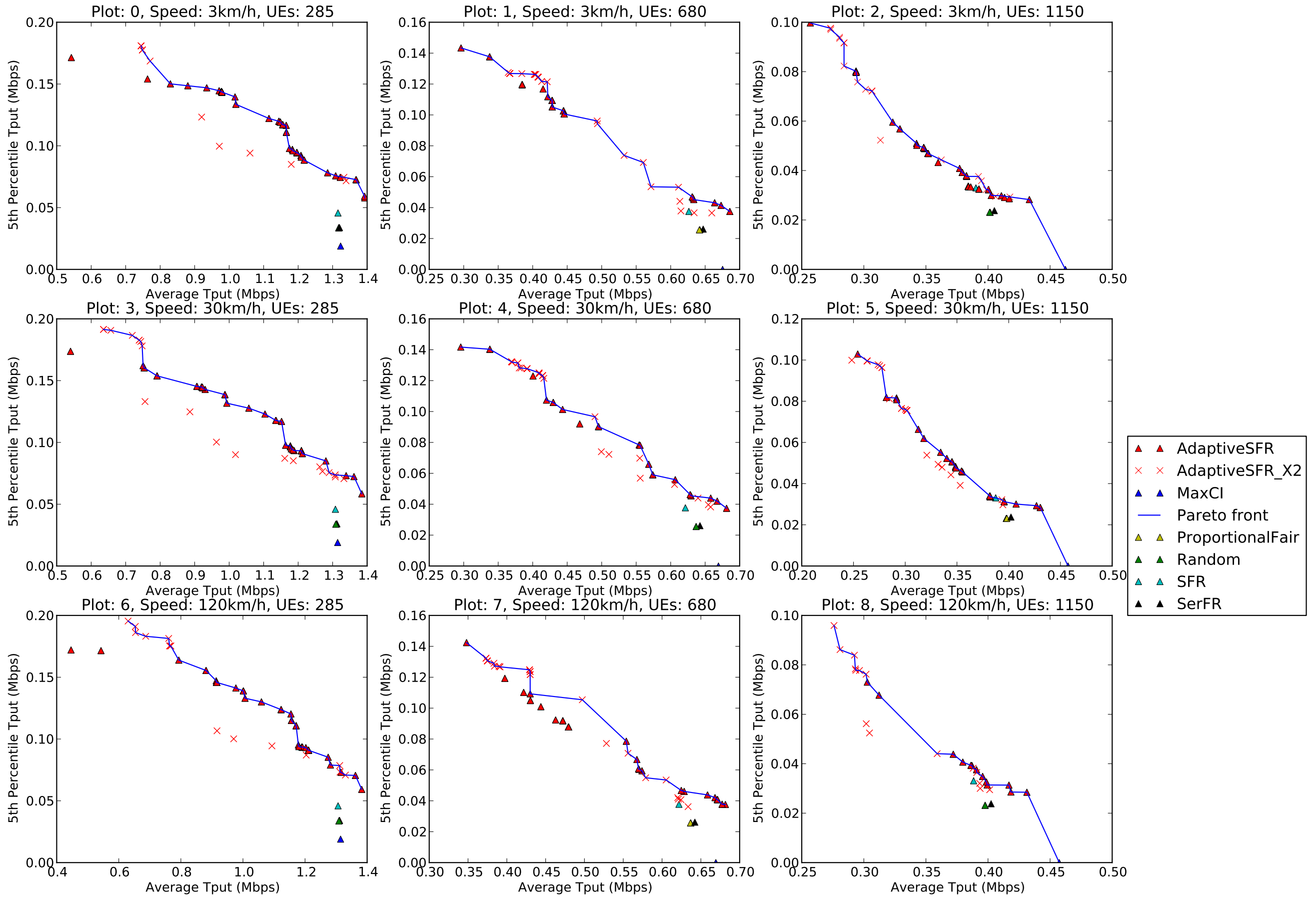


Figure 4.5: The 5th Percentile UE throughput vs average UE throughput in light traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

In this case the 5th percentile throughput is 20kbps better than aSerFR ($t(18) = 7.6648$, $p = 0.0001$). When the speed is less than 120km/h, the aSerFR+X2 is not statistically distinguishable from the aSerFR algorithm. For example in Plot 2 the aSerFR algorithm achieves a 5th percentile throughput of 100kbps while the aSerFR+X2 algorithm achieves 98kbps. The difference between these values was found to be not significant ($t(18) = 1.1630$, $p = 0.26$).

### 4.4.3 Mixed Traffic

The results of the average UE throughput can be seen in Figure 4.6. The average throughput of each algorithm under mixed traffic networks displays a similar pattern to the average throughput under light traffic loads. In particular the maximum rate algorithm again achieves an improved average UE throughput, with the aSerFR algorithm performing below the other algorithms and the aSerFR+X2 algorithm achieving the worst performance of all. The remaining algorithms performed at a very similar level.

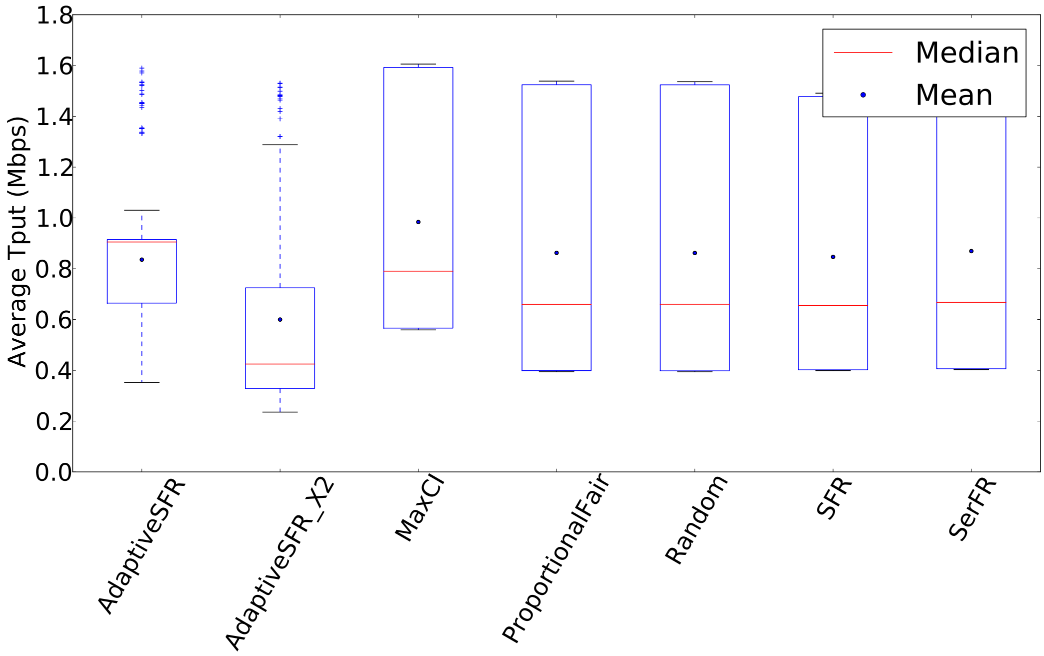


Figure 4.6: Boxplot showing the average UE throughput of each algorithm across all mixed traffic network environments.

A one-way ANOVA looking at the effect of algorithm on average UE throughput within mixed traffic network environments revealed a significant effect of algorithm

As with previous results algorithm was found to be a significant effect($F(6,1842) = 31.18$, $p = 0.001$). As suggested by Figure 4.6 the Bonferroni post-hoc comparisons showed that that the aSerFR+X2 algorithm performed significantly worse than all the other algorithms with an average mean difference of 276kbps (all $p \leq 0.001$). The aSerFR algorithm performed significantly better than the aSerFR+X2 with a mean difference of 236kbps ($p \leq 0.001$) and significantly worse than the maximum rate algorithm with a mean difference of 148kbps ($p = 0.002$). No other differences were significant.

The results in terms of 5th percentile UE throughput can be seen in Figure 4.7. The best performing algorithm in this case was the aSerFR followed by the aSerFR+X2 algorithm. The remaining algorithms performed significantly worse in each case. The performance achieved by the aSerFR and aSerFR+X2 algorithms are at least twice as good as any of the baseline algorithms.

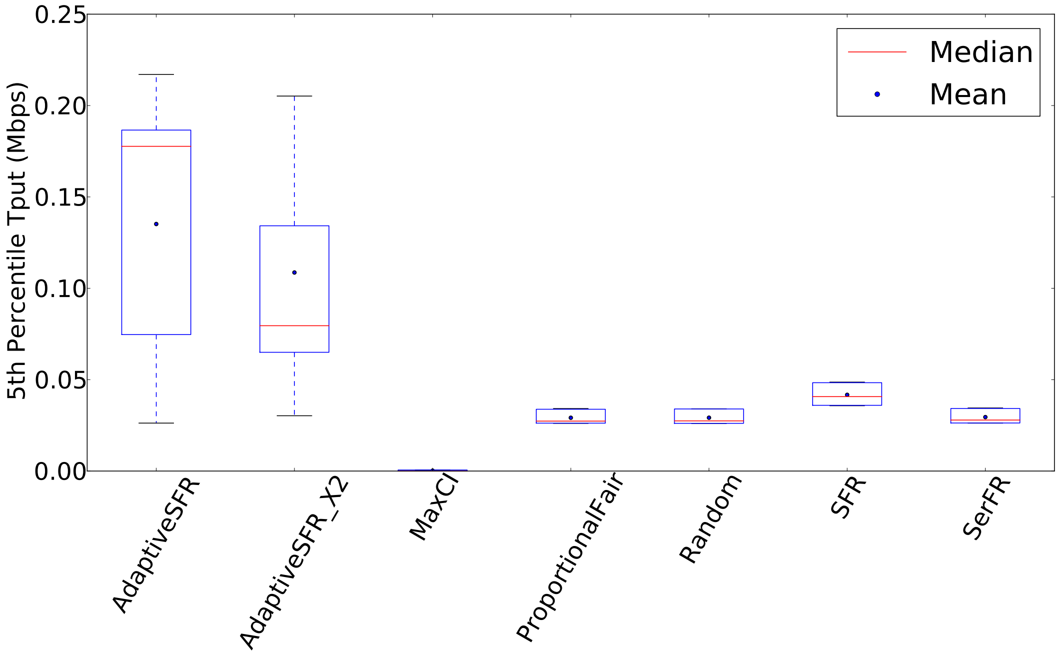


Figure 4.7: Boxplot showing the average 5th percentile UE throughput of each algorithm across all mixed traffic network environments.

As with the average throughput results algorithm was again found to be a significant effect ($F(6,1835) = 239.82$, $p \leq 0.001$). As suggested by Figure 4.7 the Bonferroni post-hoc comparisons showed that the aSerFR outperformed each of the other algorithms, with average mean difference of 96kbps (all $p \leq 0.001$). The difference in performance between the aSerFR and the aSerFR+X2 was a significant decrease of 27kbps ($p \leq 0.001$). While the aSerFR+X2 algorithm was outperformed by the aSerFR algorithm it in turn outperformed the remaining algorithms by an average mean difference of 82kbps ($p \leq 0.001$).

The results for each algorithm under each network environment with mixed traffic can be seen in Figure 4.8 which shows that the aSerFR+X2 algorithm is often dominated by the aSerFR algorithm such that it aSerFR generally performs best in terms of both 5th percentile throughput and average throughput and that aSerFR+X2 offers no performance improvement. The exception to this case is when there are 1150 UEs on the network. In these cases the aSerFR+X2 algorithm achieves significantly

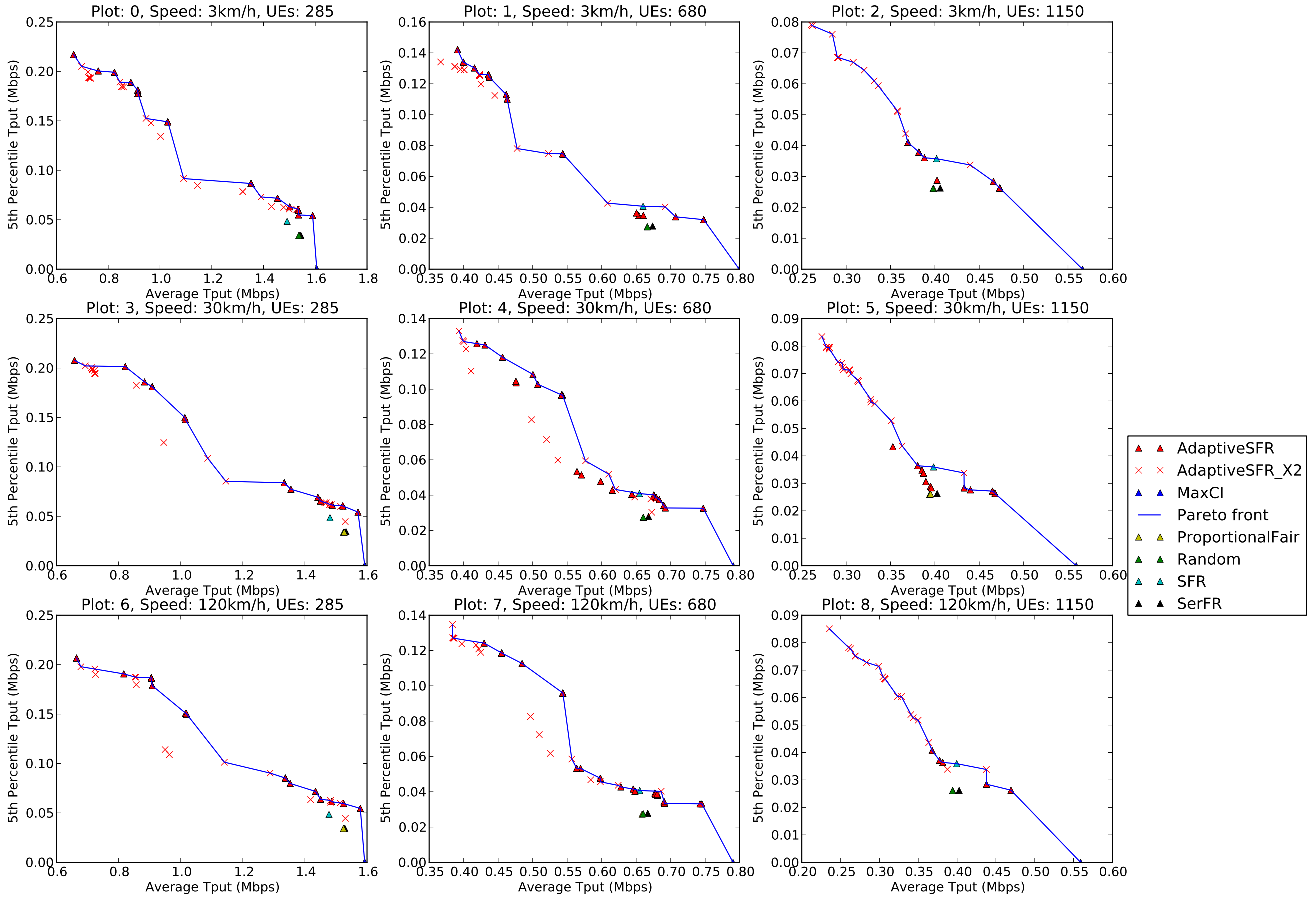


Figure 4.8: The 5th Percentile UE throughput vs average UE throughput in mixed traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

improved 5th percentile throughput.

When the speed of the 680 users is 3km/h the aSerFR+X2 algorithm generally achieves a very similar level of performance to the aSerFR algorithm. When the speed is greater than 3km/h the aSerFR+X2 does improve the 5th percentile throughput by 10kbps with an associated reduction in average UE throughput of 40kbps.

When 1150 users are present on the network the aSerFR+X2 algorithm achieves better 5th percentile throughput, improving on aSerFR by at least 40kbps. For example in Plot 8 the aSerFR+X2 algorithm achieves a 5th percentile throughput of 85kbps which is significantly better than the 41kbps achieved by the aSerFR (t(18) = 28.4917, p = 0.0001).

It can also be seen that the results produced by the aSerFR+X2 algorithm generally feature heavily in the Pareto front in particular as the number of UEs increases. The only baseline algorithm to contribute to defining the Pareto front was the SFR algorithm.

### 4.4.4 Heavy Traffic

The result for the average UE throughput under Heavy traffic can be seen in Figure 4.9. It can be seen that the maximum rate algorithm achieves the highest performance in terms of the average UE throughput by approximately 600kbps. It can also be seen that the aSerFR+X2 algorithm achieves better performance than the aSerFR by over 100kbps.

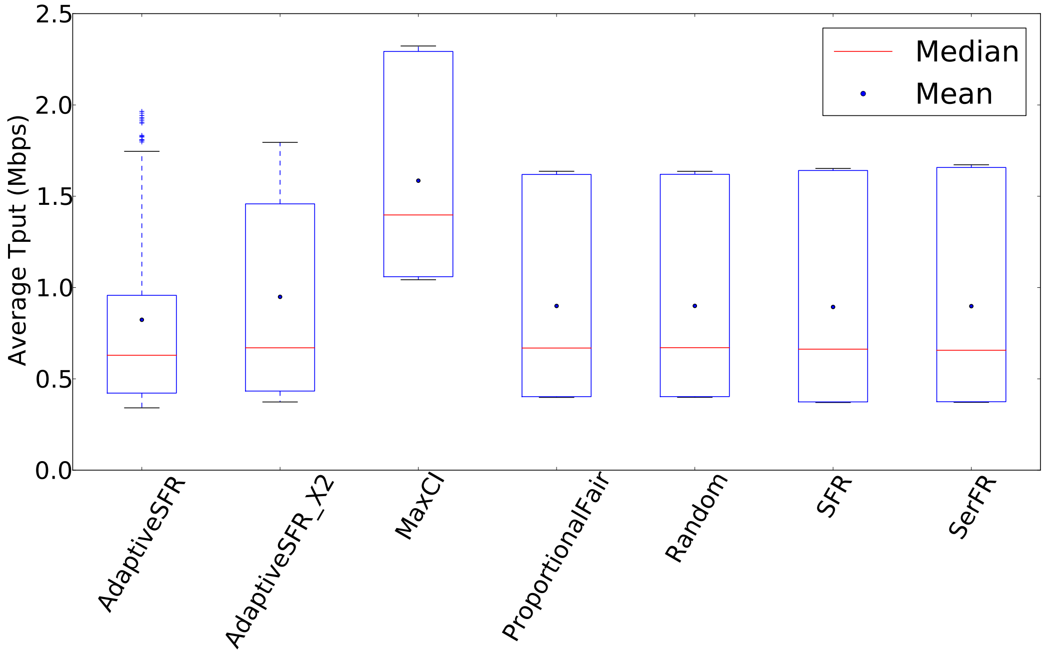


Figure 4.9: Boxplot showing the average UE throughput of each algorithm across all heavy traffic network environments.

As with the other traffic levels algorithm proved to be a significant effect (F(6,1461) = 29.81, $p \leq 0.001$). As suggested by Figure 4.9 the Bonferroni post-hoc comparisons showed that the maximum rate algorithm performed significantly better than each of the other algorithms (average mean difference = 692kbps, all $p \leq 0.001$). The aSerFR+X2 algorithm while performing significantly worse than the maximum rate algorithm, with a mean difference of 636kbps ($p \leq 0.001$), did provide a significant improvement when compared to the aSerFR algorithm with a mean difference of 126kbps ($p = 0.01$). No other differences were significant.

Figure 4.10 shows the 5th percentile UE throughput results. In the previous chapter it was found that the aSerFR algorithm did not perform as well as the baseline algorithms under heavy traffic. However with the addition of the X2 protocol the aSerFR+X2 algorithm performs best in terms of 5th percentile throughput.

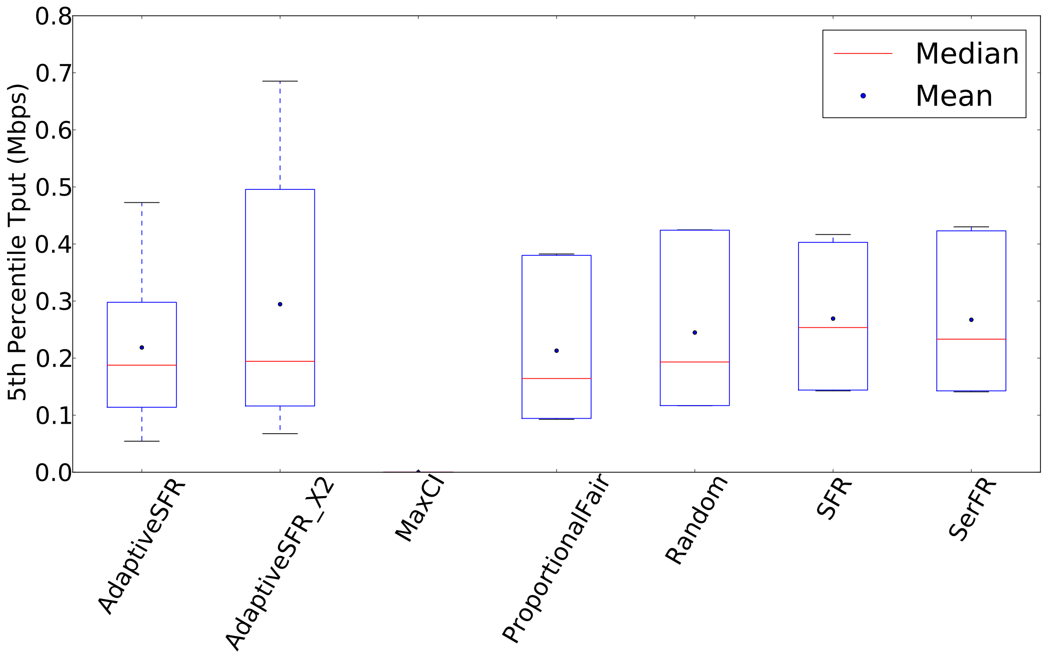


Figure 4.10: Boxplot showing the average 5th percentile UE throughput of each algorithm across all heavy traffic network environments.

A one-way ANOVA looking at the effect of algorithm on 5th percentile UE throughput within heavy traffic network environments revealed a significant effect of

Algorithm was again found to be a significant effect ($F(6,1461) = 58.68$, $p \leq 0.001$). As suggested by Figure 4.10 the Bonferroni post-hoc comparisons showed that the aSerFR algorithm performed worse than the SFR and SerFR algorithms with an average mean difference of 50kbps ($p = 0.02$). The aSerFR+X2 algorithm in comparison is statistically indistinguishable from the SFR and SerFR algorithms and performs significantly better than the remaining algorithms with an average mean difference of 125kbps ($p \leq 0.05$).

Under heavy traffic conditions, as can be seen in the individual results in Figure 4.11, the results are somewhat varied. When there are only 285 UEs on the network the aSerFR+X2 algorithm clearly performs the best in terms of 5th percentile throughput, providing an improvement of 200kbps over the aSerFR algorithm with an associated reduction in average UE throughput of 100kbps.

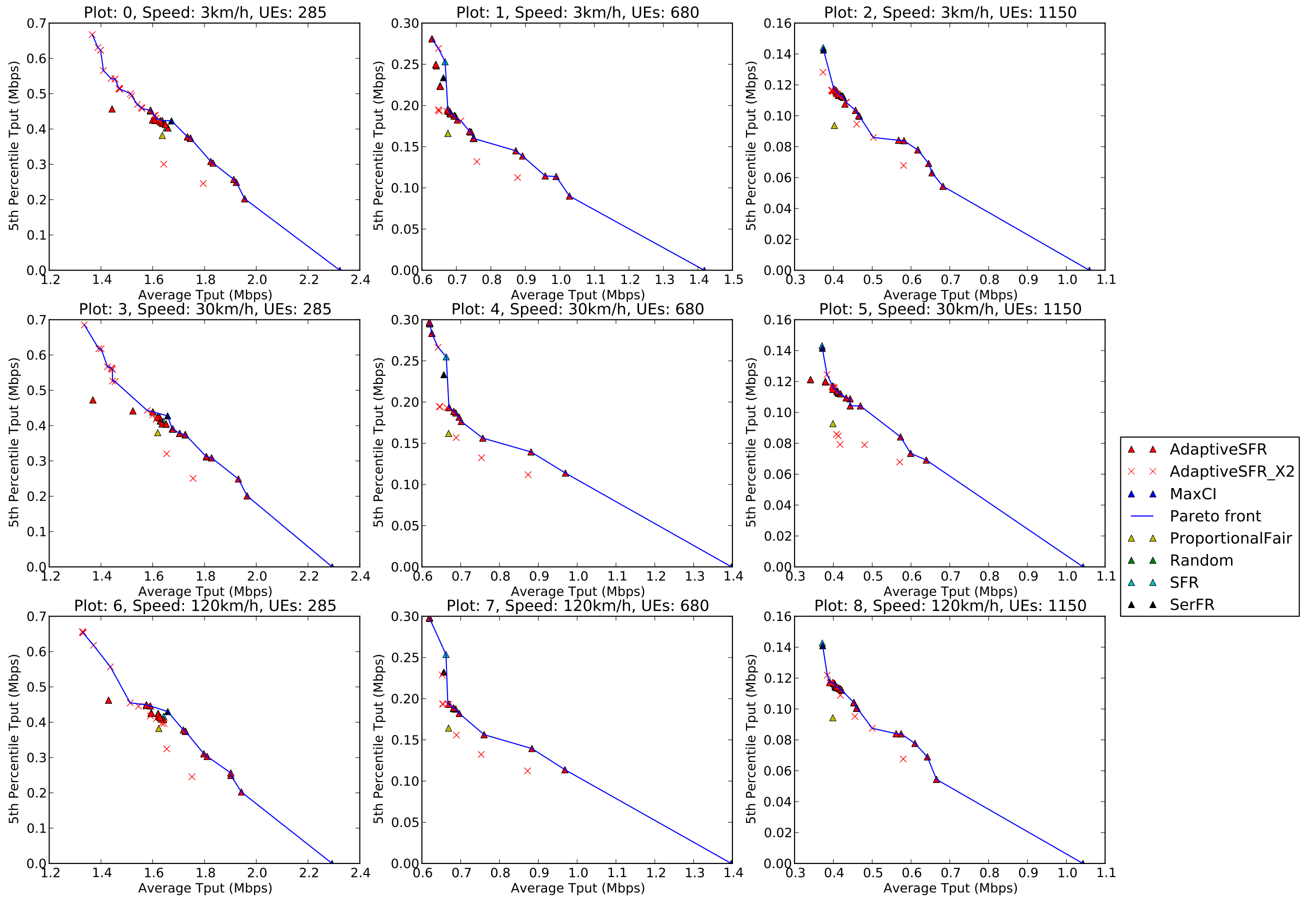


Figure 4.11: The 5th Percentile UE throughput vs average UE throughput in heavy traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

In comparison when there are 680 or 1150 UEs on the network the aSerFR+X2 algorithm is largely dominated by the aSerFR algorithm and offer no performance improvements. This shows that the improvements offered by the aSerFR+X2 algorithm, at least under heavy traffic loads, are due in most part to its performance under a smaller number of UEs.

In network environments under heavy traffic load the aSerFR+X2 algorithm contributed significantly to the Pareto front along with the aSerFR, SFR and SerFR algorithms.

### 4.4.5 Full Traffic

The results under Full traffic can be seen in Figure 4.12. As can be seen from the average UE throughput results the maximum rate algorithm outperforms the other algorithms by a very large margin, over 100% or 1Mbps. As under heavy traffic conditions the aSerFR+X2 algorithm provides some improvement in performance over the aSerFR algorithm.

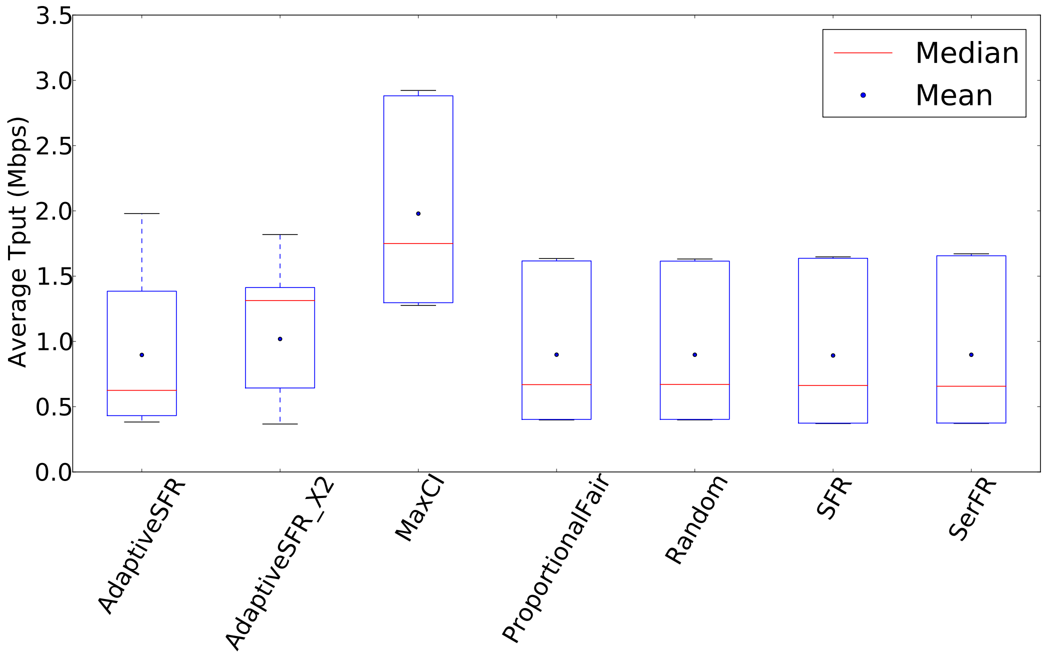


Figure 4.12: Boxplot showing the average UE throughput of each algorithm across all full traffic network environments.

A significant effect of algorithm was discovered ($F(6,1590) = 61.96$, $p \leq 0.001$). As suggested by Figure 4.12 the Bonferroni post-hoc comparisons showed that the maximum rate algorithm performed significantly better than each of the other algorithms (average mean difference = 1.06Mbps, all $p \leq 0.001$). As suggested by the figure the aSerFR+X2 improved upon the performance of the SFR algorithm with a mean difference of 122kbps ($p = 0.005$). All the remaining algorithms are statistically indistinguishable in terms of their average UE throughput.

The 5th percentile throughput results, seen in Figure 4.13 show that the aSerFR+X2 algorithm provides large performance gains over the other algorithms. The results in the previous chapter showed that the aSerFR offered no improvement on the existing algorithms. This shows that the addition of the X2 protocol enhances its performance.

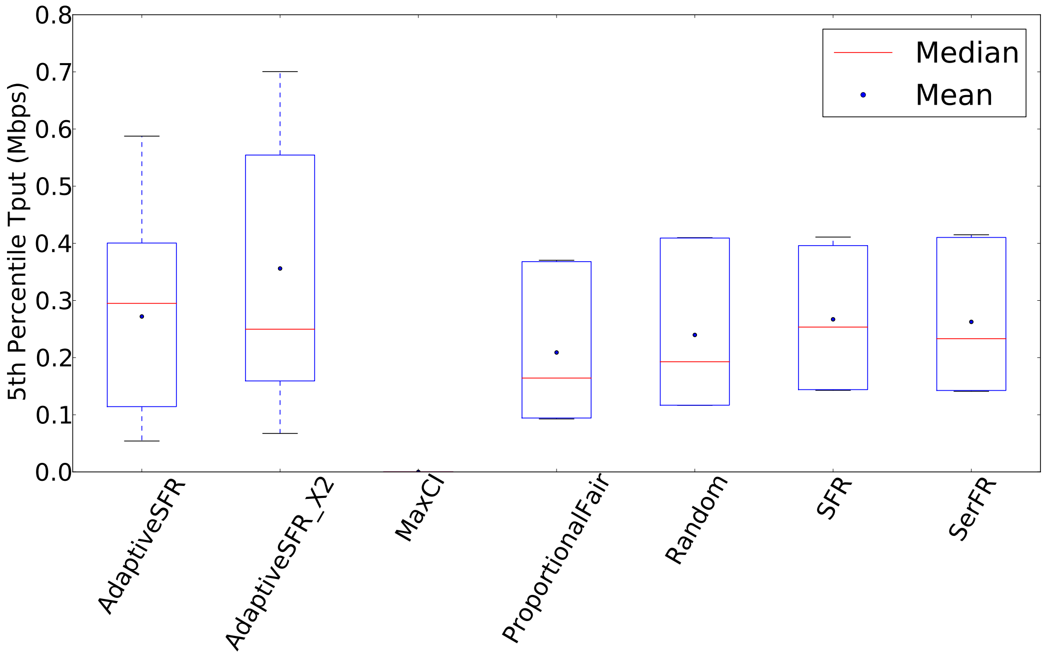


Figure 4.13: Boxplot showing the average 5th percentile UE throughput of each algorithm across all full traffic network environments.

Again a significant effect of algorithm was found (F(6,1590) = 65.92, $p \leq 0.001$). As suggested by Figure 4.13 the aSerFR+X2 provided the best performance. The Bonferroni post-hoc comparisons showed that the aSerFR+X2 was significantly better than all the other algorithms with an average mean difference of 147kbps (all $p \leq 0.001$).

The individual results from the experiment when run with the full traffic profile can be seen in Figure 4.14. It can be seen that then there are only 285 UEs on the network the aSerFR+X2 algorithm generally performs significantly better than the aSerFR, except in the case where the speed is 30km/h. When the speed is 3km/h (Plot 0) the aSerFR+X2 scheme improves the 5th percentile throughput of the aSerFR algorithm by a 250kbps (t(18) = 14.6331, p = 0.0001), with a reduction in average UE throughput of 300kbps. When the speed is 120km/h (Plot 6) the aSerFR+X2 scheme again improves the 5th percentile throughput by 200kbps ($t(18) = 13.1272$, $p = 0.0001$) with a reduction in average throughput of 60kbps.

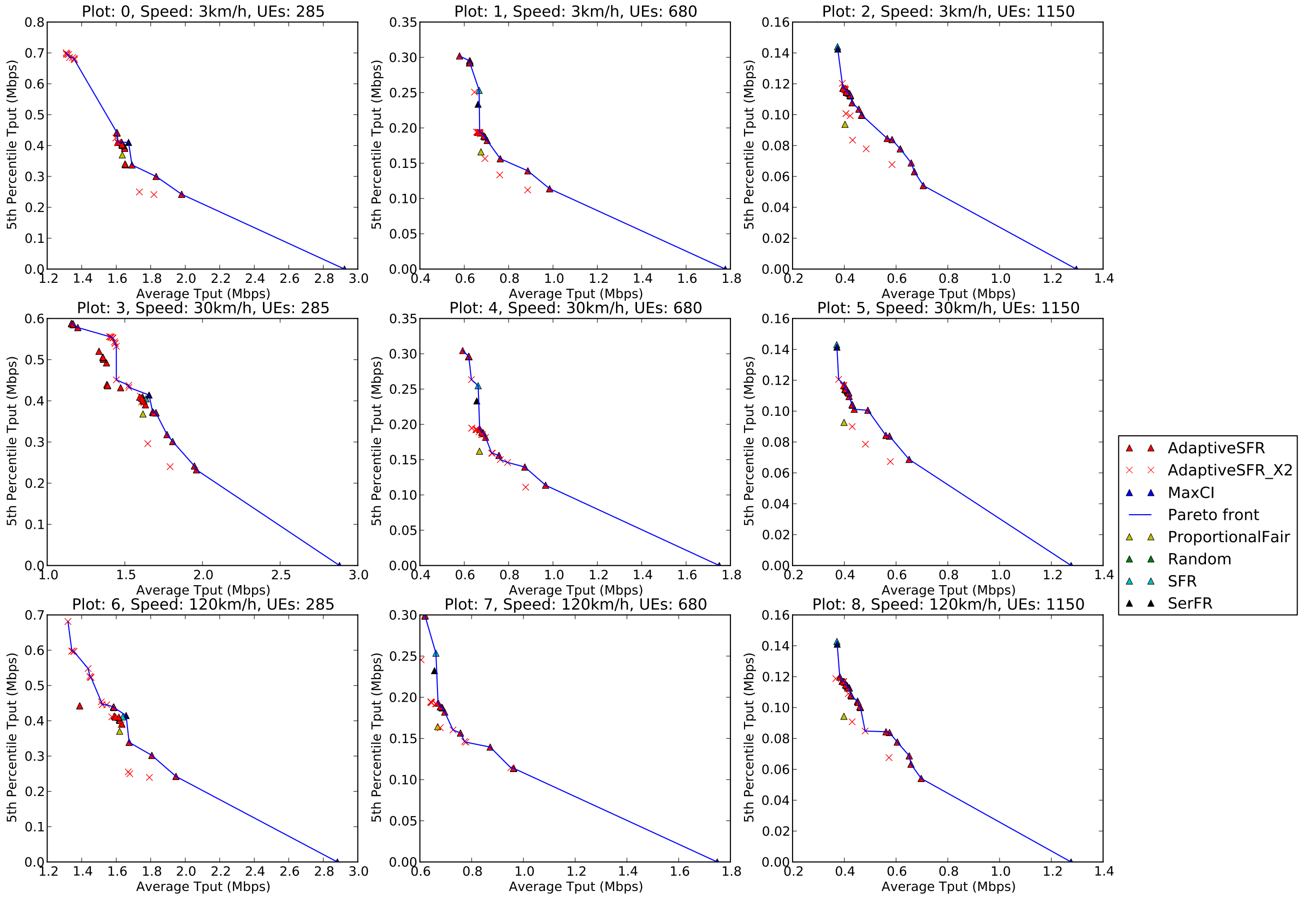


Figure 4.14: The 5th Percentile UE throughput vs average UE throughput in full traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

When there are 680 or 1150 UEs on the network the aSerFR+X2 scheme does not improve the performance of the aSerFR algorithm.

It can also be seen from these results that the aSerFR+X2 algorithm contributes to the Pareto front in each case. Often, however, the results obtained from the aSerFR+X2 algorithm are dominated by others, in particular those produced by the aSerFR algorithm. As discussed under heavy traffic the aSerFR+X2 offers the greatest performance improvements when there are smaller numbers of UEs on the network.

### 4.4.6 Summary

As reflected by the overall analysis of the results, the aSerFR+X2 algorithm was able to improve significantly on the performance of both the aSerFR and baseline algorithms in terms of 5th percentile UE throughput. In every traffic level, with the exception of heavy traffic, the aSerFR+X2 algorithm outperformed the baseline algorithms by a significant margin. When the traffic was heavy the aSerFR+X2 algorithm was able to match the best performing of the baseline algorithms. The improvements were particularly pronounced under the light and full traffic levels with the aSerFR+X2 algorithm performing on average 147kbps better than aSerFR under full traffic conditions.

## 4.5 Discussion

This chapter introduced a new ICIC scheme that built on the aSerFR scheme presented in the previous chapter and incorporated the X2 link. This scheme was presented as an example of networked organisation, that is using the X2 link to allow direct communication and information sharing between eNBs. A number of experiments were described which compared the performance of the aSerFR+X2 scheme with the aSerFR and other schemes in order to see if this form of networked organisation offered any performance improvements.

The results described in the previous section demonstrate that in general the networked approach can provide significant improvements in 5th percentile throughput.

Improved results were achieved under light, mixed and full traffic loads indicating that this approach is perhaps less dependent on the resource constraints of the network that the aSerFR was found to be. In the one case where the aSerFR+X2 scheme was not able to exceed the results produced by the existing ICIC schemes it was able to match their 5th percentile UE throughput.

It was also observed that when the aSerFR+X2 scheme did not improve the 5th percentile throughput it was generally dominated by the results of the aSerFR and offered no improvement in terms of either 5th percentile throughput or average throughput. However it was noted that in those cases when the aSerFR+X2 scheme did increase the 5th percentile throughput achieved by the network the gains were generally very significant, in some cases doubling the best performance achieved by the other algorithms.

The results imply that this form of networked organisation can be used to significantly improve the 5th percentile UE throughput of the network. It is also implied that networked organisation of this type is not as limited by the contention on the radio resources of the network as the local organisation approach was found to be.

# Chapter 5

# De-centralised Organisation

Previous chapters have described a number of Inter-Cell Interference Coordination (ICIC) schemes, both existing and new, which are designed to reduce the level of interference in Long Term Evolution (LTE) networks. The new ICIC schemes developed as part of this research have been developed around the idea of localised organisation (Adaptive Softer Frequency Reuse (aSerFR)) and networked organisation (Adaptive Softer Frequency Reuse + X2 (aSerFR+X2)). Local organisation functions at the level of the individual network nodes, Evolved NodeBs (eNBs), without any communication between nodes or knowledge of neighbouring nodes. Networked organisation made use of the X2 link available in LTE to enable communication and knowledge sharing between eNBs. It was demonstrated through a series of experiments that these approaches could provide significantly improved network performance over existing ICIC schemes in the literature.

This chapter introduces the concept of de-centralised ICIC schemes. A de-centralised ICIC scheme is one where the process of computing the optimisation takes place on the Radio Access Network (RAN) itself, that is the eNBs themselves are used to perform the optimisation calculation.

In the previous ICIC schemes, all of which could be described as centralised schemes, the optimisation process takes place on some central network node, not itself involved in the operation of the RAN. This centralised node is responsible for calculating the optimal network configuration to be used by the eNBs. These configuration parameters were then applied uniformly to all the eNBs on the network.

A de-centralised approach, facilitated by the X2 link, allows the optimisation itself to be performed by the eNBs themselves in real-time.

### 5.0.1 Central and Distributed Organisation

The optimisation schemes discussed in previous chapters can all be described as centralised Self-Organising/Optimizing Network (SON) systems. A centralised SON is an approach to optimisation where the optimisation process itself takes place in an offline fashion [2]. That is the computation required to determine the optimal network configuration takes place on a node that is not directly involved with the running of the network. This network node then pushes the calculated configuration to those nodes that are involved in the running of the network, i.e. the eNBs, so that they may make use of the calculated optimisation.

There are several potential problems with this approach. Firstly the parameters used by each eNB are the same across the network; this means that even though eNBs will be experiencing different levels of load and User Equipment (UE) behaviour they will all have the same scheduling parameters as determined by the optimisation. Such a scheme is unlikely to be optimal since it is much more likely that individually tailoring the parameters of each eNB to its own network environment will lead to improved performance as the parameters will better reflect the reality of the network.

Secondly such a centralised scheme is not very responsive to changes in the network environment. For example networks typically see different usage levels and patterns at different times during the day and one set of parameters is unlikely to be optimal for each condition. In order to adapt to the changing environment the optimisation would need to be re-run when the network environment changes. This means there will be an inevitable disparity between the parameters used by the eNBs and the network environment they are operating in.

An alternative to the centralised approach is to use a de-centralised or distributed scheme. Rather than running the optimisation on a central network node, it is run over the network elements themselves. This means that the eNBs can optimise themselves and adapt to the network changes that they experience in a much more responsive way. It also means that the parameters obtained by the optimisation are tailored specifically to the eNB which is running the optimisation. It does however mean that the algorithm must be computationally cheap to run since the eNBs have little in the way of spare computational capacity.

### 5.0.2 A De-centralised ICIC Algorithm

The de-centralised ICIC scheme presented in this section, known as De-centralised Softer Frequency Reuse (dSerFR), is based on those described in previous chapters, specifically aSerFR. aSerFR provided a parameterised scheduler which could be optimised for different network environments with the use of a Genetic Algorithm (GA). However, rather than running this GA as a centralised process dSerFR runs the GA on the eNBs themselves.

In the previously described ICIC schemes the GA is run on a central process. This process stores a population of chromosomes which represent different configurations of the scheduler. This is mutated over time and evaluated using the network simulator described in Section 3.2. In dSerFR there is no central process to run the GA but instead each eNB maintains a single chromosome which represents its own current scheduler configuration. At start-up this configuration is initialised with random values. The eNB itself then mutates this configuration over time while measuring the effect it has on its own, and possibly its neighbours, performance. This allows each eNB to independently evolve their own set of scheduling parameters which are tailored to the specific network environment in which they are located.

In order for this approach to be effective it is important that all of the eNBs on the network do not attempt to mutate their chromosomes and measure their performance change at the same time. If this were to be allowed to happen it would be difficult to determine whether the performance change of an eNB was due to its own parameters changing or those of its neighbours. Ideally while an eNB is mutating its chromosome and measuring the change in its performance all other variables would be kept constant. In reality this is not possible since the network environment will always be changing around the eNBs as UEs leave and join cells, initiate calls and initiate data sessions. However it is still important that the algorithm not conflate this further by mutating multiple neighbouring eNBs at the same time.

In order to avoid this the eNBs are conceptually divided into clusters called neighbourhoods. Within each neighbourhood only one eNB is able to evolve its configuration at any one time. Each eNB periodically communicates with those eNB in its neighbourhood using the X2 link to determine whether any of them is currently mutating. If any of the other eNBs in the neighbourhood are mutating then none of the other eNBs will begin a mutation, but will wait for a period of time before querying their neighbours' status again. However if none of the eNBs are mutating then the eNB which began the query will begin to evolve.

There are two different mutation strategies used by dSerFR: greedy and consensus based. The mutation operator used is in both cases is the basic single point mutation as described in Section 2.5.1.

**Greedy Selection** Prior to mutation the eNB stores the value of the current 5th percentile throughput of the UEs that it is serving as a fitness benchmark. One of the genes in the chromosome, a single configuration value, is chosen randomly and its value is mutated. The eNB then continues serving its UEs for some time window using this mutated configuration. After this window has passed the eNB again takes a measure of the 5th percentile throughput of its UEs and compares that value with the value measured at the beginning of the mutation. If this fitness value has improved over this period then mutation is kept, if not it is discarded and the configuration is reverted.

**Consensus Selection** The consensus selection strategy differs from the greedy strategy mainly in that not only does the mutating eNB record its changing fitness during the mutation but all the eNBs in the neighbourhood record the changes to their fitness. When an eNB begins to mutate it alerts those eNBs in its neighbourhood using the X2 link so that all the eNBs take a measure of their current 5th percentile throughput. The mutating eNB then mutates its configuration and begins to use this mutated configuration to serve its UEs. Once the mutation window has passed all the eNBs in the neighbourhood measure their 5th percentile throughput a second time and communicate to the mutating eNB whether their fitness has increased or decreased. The mutating eNB will then take these responses into account to determine whether it should accept the new configuration value. A parameter of the dSerFR algorithm specifies what proportion of the neighbourhood must have seen an increase in their fitness value before the mutating eNB can accept the new configuration. If the proportion of eNBs in the neighbourhood that observed an increase in their fitness value is less than the dSerFR parameter, meaning that a number of the eNBs saw a decrease in their performance, then the mutating eNB will discard the mutation as it has not been beneficial to the neighbourhood.

This scheme is computationally simple. Each eNB needs only keep track of its own fitness over a time window along with some basic information about its chromosome. Currently however this algorithm will only optimise a single objective function, in this case 5th percentile throughput. It would be possible to optimise in a multi-objective way but care would have to be taken to develop a function to determine the fitness across competing objectives.

## 5.1 Experiments

In order to determine the performance of this de-centralised approach a number of experiments were undertaken and the results compared to the approaches described in the previous chapters.

In contrast to previous simulation runs the simulator was run for 200,000 iterations which represents approximately 5 seconds of real network time. This is important since the performance of the optimisation changes over time. However it is important to ensure that this does not provide this approach with a benefit not given to the previous algorithms. A pilot study was carried out to determine whether this extended run-time would confer any advantage to the dSerFR algorithms. The graphs shown in Figures 5.1 and 5.2 show that each of the existing algorithms generally stabilises after approximately 100 iterations while the dSerFR algorithm continues to change, in terms of both 5th percentile UE throughput and average UE throughput.

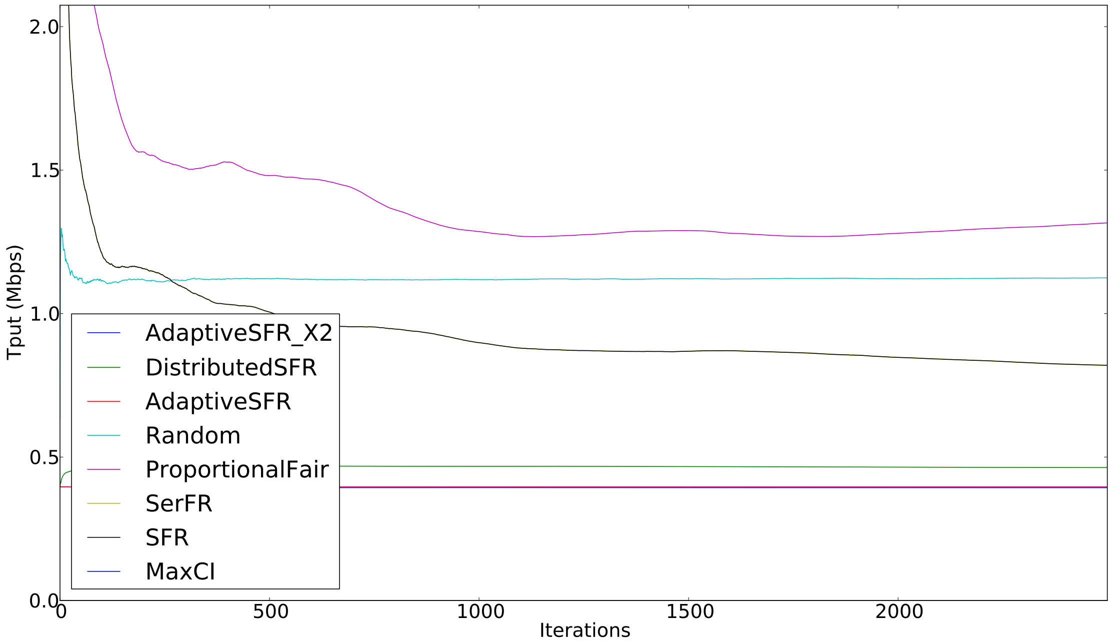


Figure 5.1: The convergence of the average UE throughput values in the simulator over time.

The parameters used for the dSerFR scheme can be seen in Table 5.1.

| Parameter | Value |
|---|---|
| Mutation Window | 100 |
| Consensus Proportion | 0.75 |
| Mutation Strength | 1 |
| Mutation Operator | Point Mutation |

Table 5.1: The parameters used for the dSerFR.

## 5.2 Results

The results of this experiment can be seen in Figures 5.7 to 5.16. Note that for the sake of clarity only the best performing algorithms from the previous chapters have been shown.

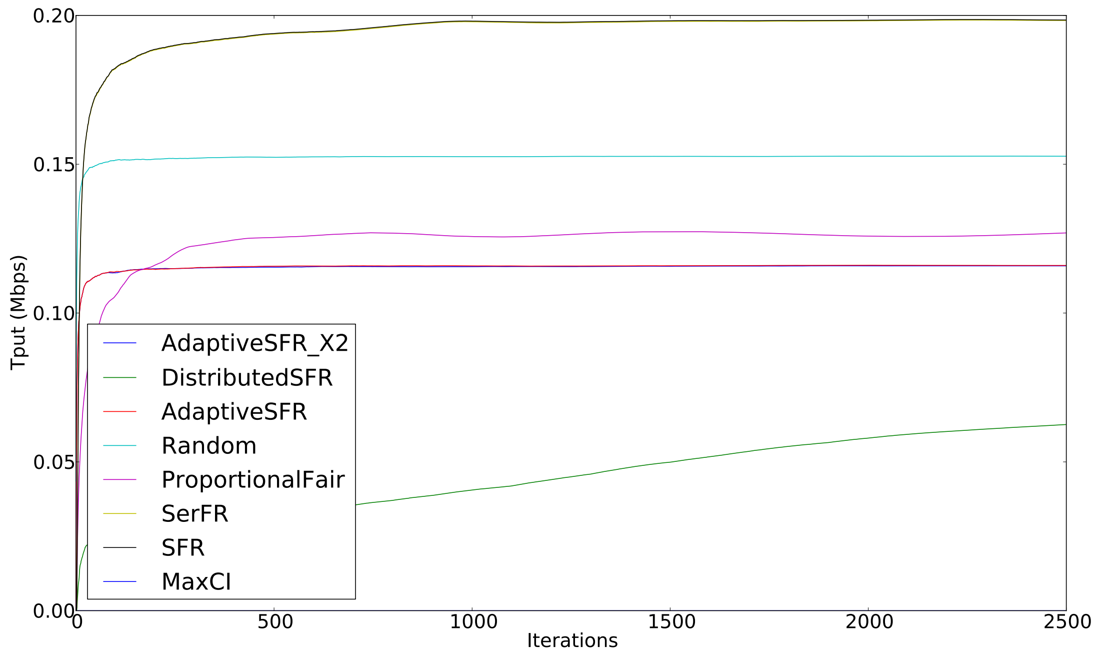


Figure 5.2: The convergence of the 5th percentile UE throughput values in the simulator over time.

### 5.2.1 Overall Analysis

As in previous chapters an ANOVA for average UE throughput was run which demonstrated a significant main effect for algorithm, (F(8,7258) = 1036.65, $p \leq 0.001$). Bonferroni analysis were then undertaken to compare the results of each algorithm taking into account the fact that multiple comparisons have taken place.

As suggested by Figure 5.3 the maximum rate algorithm achieved a significantly higher average UE throughput when compared to all the other algorithms with an average mean difference between the maximum rate algorithm and each of the remaining algorithms of 492kbps (all $p \leq 0.001$). The next best performing algorithms were dSerFR algorithms, both greedy and consensus based, followed closely by the baseline algorithms. The aSerFR and aSerFR+X2 again performed significantly worse than all the other algorithms in terms of average UE throughput. Of particular interest in the context of this chapter the dSerFR algorithms, both the greedy and consensus based, performed significantly better than both the aSerFR and aSerFR+X2 (all $p \leq 0.001$).

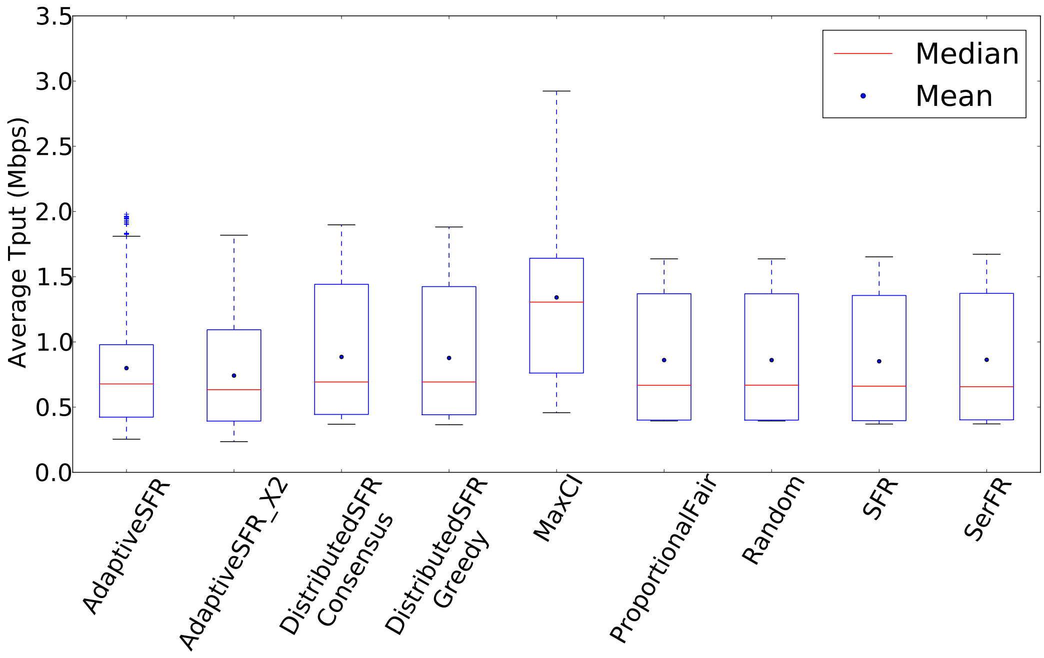


Figure 5.3: Boxplot showing the average UE throughput of each algorithm across all traffic levels.

As in previous analyses a number of interaction effects were discovered. In particular the strongest interaction was between algorithm and traffic level (F(24,7258) = 181.96, $p \leq 0.001$) which called for follow-up simple effects analyses within each level of traffic. These subsequent analyses (ANOVAs investigating the effect of algorithm on average throughput within each traffic level) are detailed in separate sections below.

The same process was carried out to investigate the effects of algorithm and environment on 5th percentile UE throughput. This demonstrated a significant main effect for algorithm (F(8,7259) = 934.31, $p \leq 0.001$). The results for each algorithm in terms of 5th percentile UE throughput can be seen in Figure 5.4. As in the previous chapter the aSerFR and aSerFR+X2 algorithms achieved the best 5th percentile throughput. The dSerFR algorithms, while not matching the performance of the aSerFR and aSerFR+X2, were able to match the performance of the Soft Frequency Reuse (SFR) and Softer Frequency Reuse (SerFR) algorithms with an average mean difference of 2kbps (all $p = 1$).

Once again separate analyses were conducted within each traffic level in view of the significant interaction between traffic level and algorithm (F(24,7258) = 113.16,

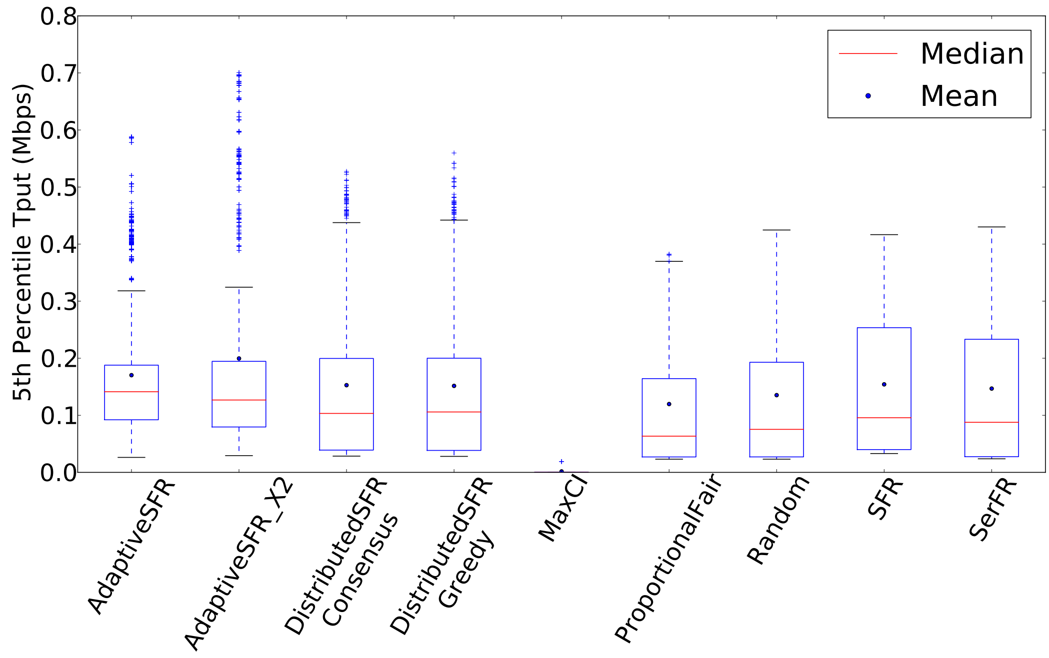


Figure 5.4: Boxplot showing the 5th percentile UE throughput of each algorithm across all traffic levels.

$p \leq 0.001$).

### 5.2.2 Light Traffic

The results in terms of average UE throughput can be seen in Figure 5.5. It can be seen from this figure that, as expected, the relative performance of each of the algorithms is very similar to that described in Chapter 4. The average throughput achieved by the dSerFR algorithms was able to match that achieved by the baseline algorithms and thus improve on that achieved by the aSerFR and aSerFR+X2.

A one-way ANOVA looking at the effect of algorithm on average UE throughput within light traffic network environments was again carried out. This revealed a significant effect of algorithm ($F(8,2132) = 17.27$, $p \leq 0.001$). As suggested by Figure 5.5 the Bonferroni post-hoc comparisons showed that the dSerFR algorithms, both the greedy and consensus based, were significantly better than the aSerFR+X2, with a mean difference of 264kbps ($p \leq 0.001$) and 258kbps ($p \leq 0.001$) respectively, and was able to match the performance of all of the baseline algorithms, including the maximum rate algorithm, with an average mean difference of 7kbps (all $p = 1.0$) and 0.5kbps (all $p = 1.0$) respectively.

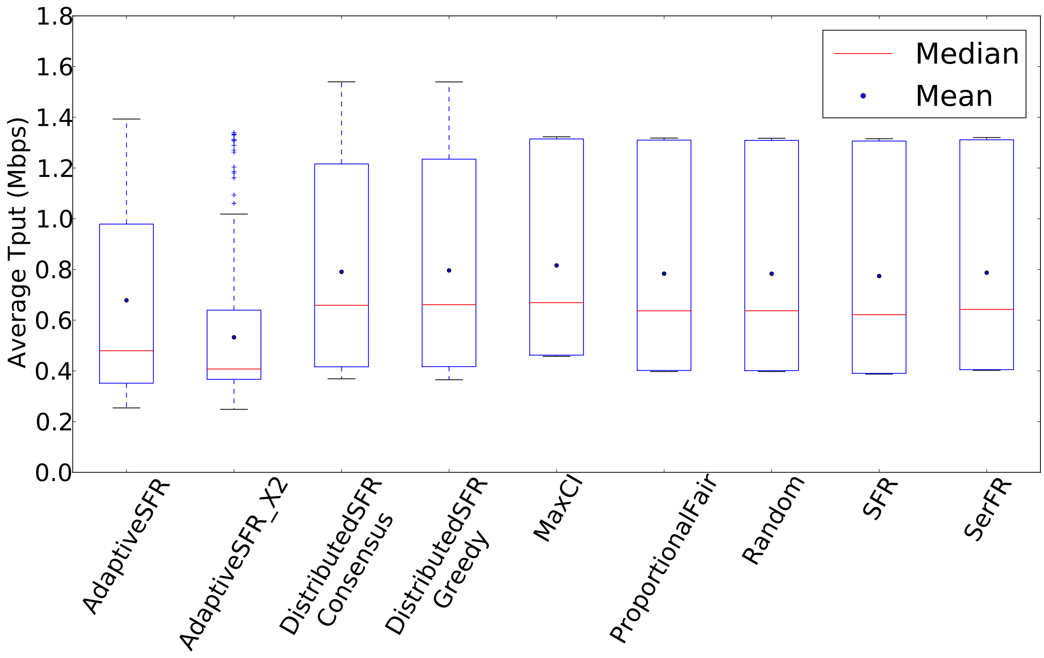


Figure 5.5: Boxplot showing the average UE throughput of each algorithm across all light traffic network environments.

Figure 5.6 shows the results of each algorithm in terms of the 5th percentile UE throughput. It can be seen that the 5th percentile throughput of the two dSerFR algorithms, while not achieving the same performance as either the aSerFR or the aSerFR+X2, do improve on the 5th percentile throughput achieved by the baseline algorithms.

The ANOVA found a significant effect of algorithm ($F(8,2132) = 205.35$, $p \leq 0.001$). As suggested by Figure 5.6 the post-hoc comparisons showed that the dSerFR algorithms, both greedy and consensus based, were significantly worse than both the aSerFR and the aSerFR+X2 with an average mean difference of 48kbps (both $p \leq 0.001$) and 48kbps (both $p \leq 0.001$) respectively. While there was no significant difference with the SFR algorithm and either of the dSerFR algorithms, the dSerFR algorithms did perform significantly better than all the other baseline algorithms with an average mean difference of 28kbps (all $p \leq 0.001$).

As in the previous chapters the results achieved by each algorithm on the light traffic network environments can be seen in Figure 5.7. This figure shows the nine different network environments run under light traffic. These are combinations of three UE density levels (285 UEs, 680 UEs, 1150 UEs) and three UE speeds (3km/h,

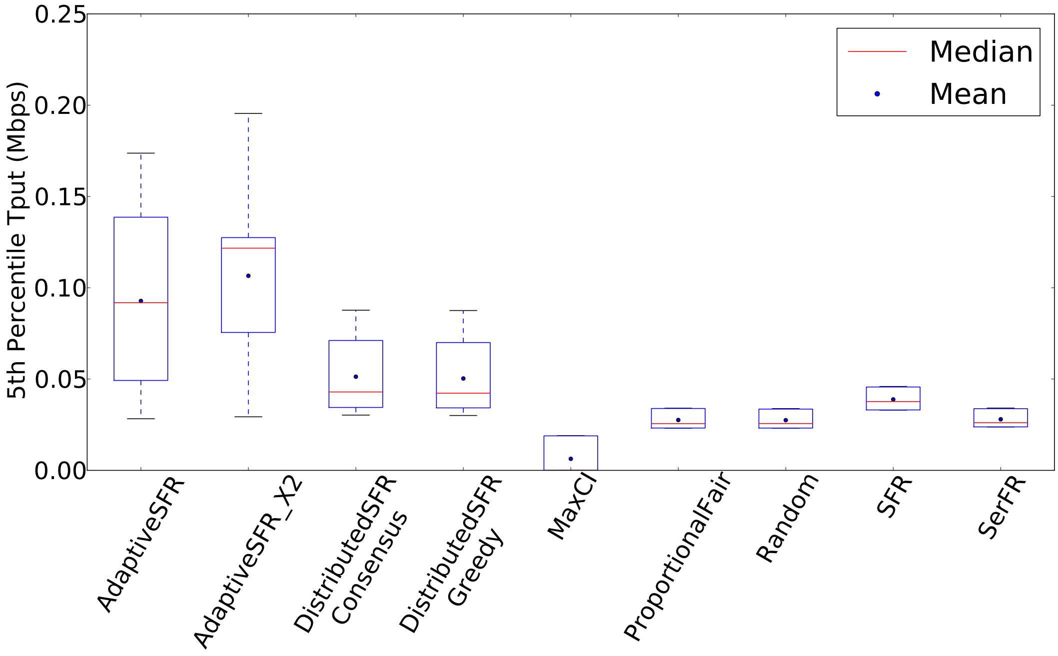


Figure 5.6: Boxplot showing the average 5th percentile UE throughput of each algorithm across all light traffic network environments.

30km/h, 120km/h).

It can be seen from Figure 5.7 that the dSerFR algorithm has mixed results under light traffic. When there are 285 or 680 UEs on the network the dSerFR algorithm achieves better 5th percentile throughput results than the baseline results. In these cases the average throughput is at least as good as the baseline algorithms.

In those cases where there are 1150 UEs on the network the dSerFR algorithm matches the performance of the SFR algorithm in terms of 5th percentile throughput, but appears to achieve an improved average throughput, however this was shown to just fall short of statistical significance. For example in Plot 8, the dSerFR algorithms achieved comparable 5th percentile UE throughput to the SFR algorithm, 33kbps in both cases. However the average UE throughput achieved by the dSerFR algorithms was 407kbps compared to the 388kbps achieved by the SFR algorithm which was found to be a statistically significant improvement ($t(18) = 2.1026$, $p = 0.0498$).

It can also be seen that in each network environment the dSerFR did not achieve better results than the aSerFR+X2, either in terms of 5th percentile throughput or average throughput. However it can be seen that in most cases the results from

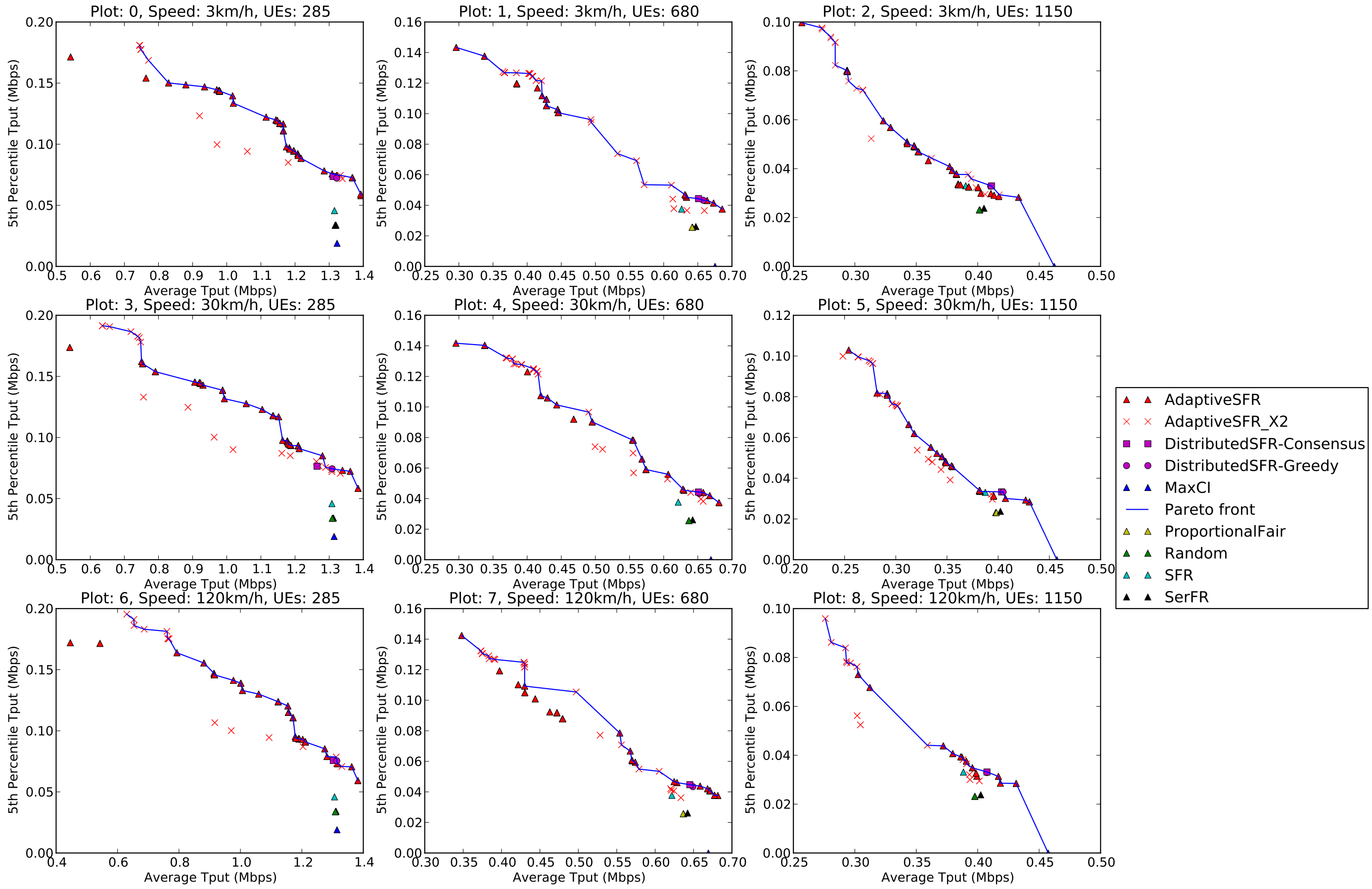


Figure 5.7: The 5th Percentile UE throughput vs average UE throughput in light traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

the dSerFR do appear as part of the Pareto front and thus are not dominated by any of the other results which shows that the results do offer an improved trade-off between the average UE throughput and the 5th percentile UE throughput.

### 5.2.3 Mixed Traffic

The results for average UE throughput can be seen in Figure 5.8. The average performance of each algorithm under mixed traffic networks displays a similar pattern to the average performance under light traffic loads. In particular the dSerFR again provide improved performance over the aSerFR and aSerFR+X2 algorithms and achieve a similar level performance to the baseline algorithms.

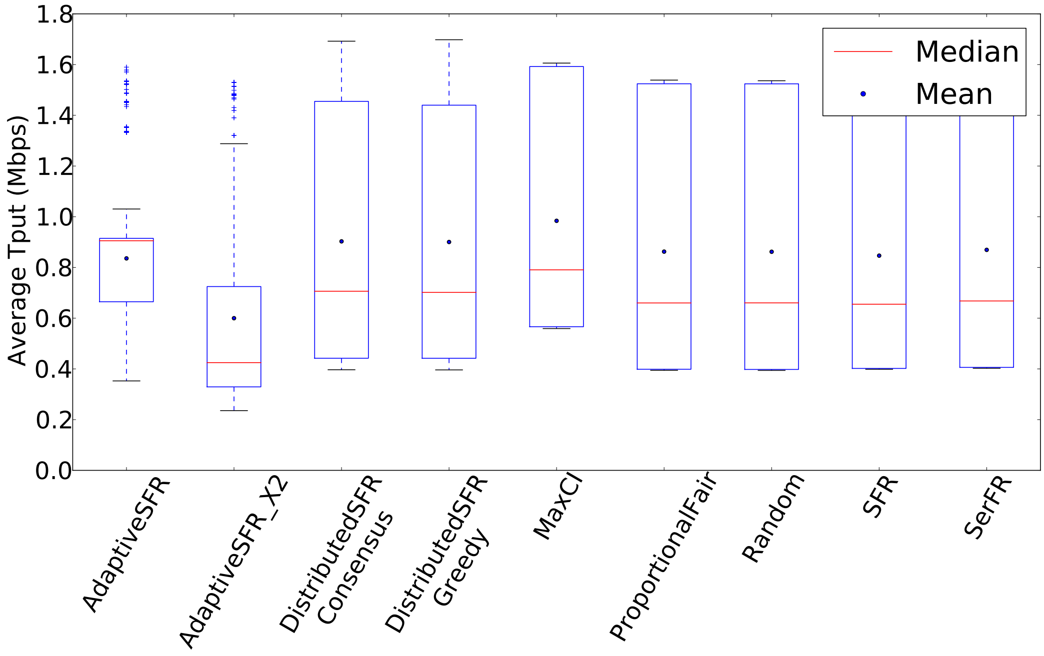


Figure 5.8: Boxplot showing the average UE throughput of each algorithm across all mixed traffic network environments.

The ANOVA again found a significant effect of algorithm (F(8,2009) = 23.50, p = 0.001). As suggested by Figure 5.8 the Bonferroni post-hoc showed that the dSerFR algorithms, both the greedy and consensus based, were significantly better than the aSerFR+X2, with a mean difference of 300kbps (p $\leq$ 0.001) and 303kbps (p $\leq$ 0.001) respectively, and was able to match the performance of all of the baseline algorithms, including the maximum rate algorithm, with an average mean difference of 15kbps (all p = 1.0) and 45kbps (all p = 1.0) respectively.

Figure 5.9 shows the average performance of each algorithm in terms of the 5th percentile UE throughput. It can be seen that in similar fashion the results found under light traffic the 5th percentile throughput of the two dSerFR algorithms, while not achieving the same performance as either the aSerFR or the aSerFR+X2, do achieve a similar level of 5th percentile throughput as the best baseline algorithm.

The ANOVA again revealed a significant effect of algorithm (F(8,2009) = 226.29, p $\leq$ 0.001). As suggested by Figure 5.9 the Bonferroni post-hoc comparisons showed that the dSerFR algorithms, both greedy and consensus based, were significantly worse than both the aSerFR and the aSerFR+X2 with an average mean difference of 81kbps (both p $\leq$ 0.001) and 80kbps (both p $\leq$ 0.001) respectively. There was

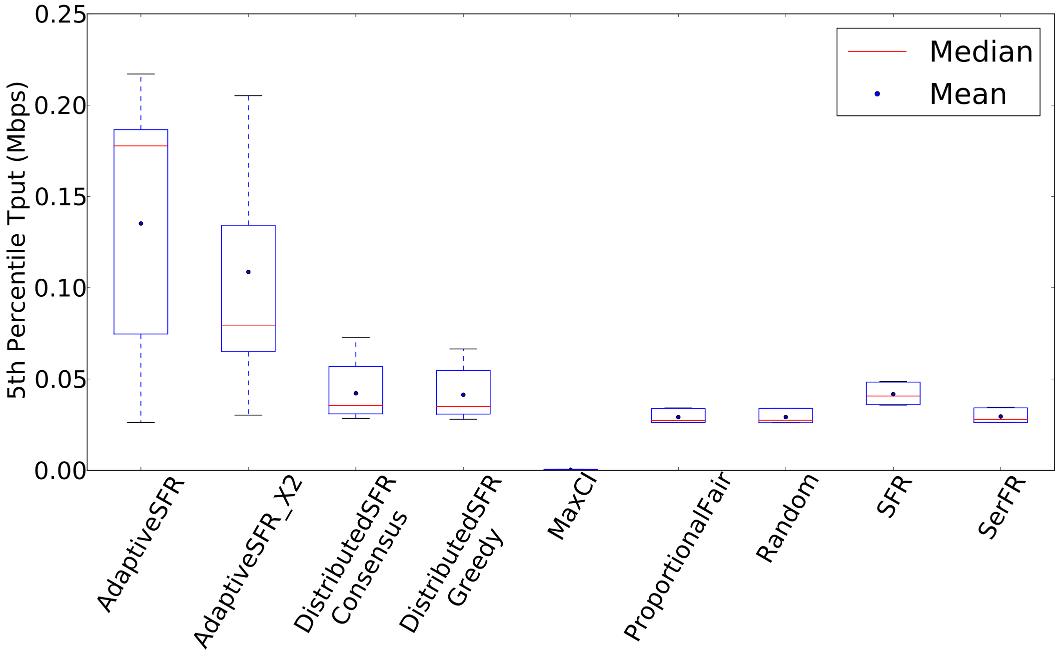


Figure 5.9: Boxplot showing the average 5th percentile UE throughput of each algorithm across all mixed traffic network environments.

no significant difference between the dSerFR algorithms and any of the baseline algorithms, apart from the maximum rate algorithm which as expect performed significantly worse than all of the other algorithms in terms of the 5th percentile UE throughput, with an average mean difference of 56kbps (all p $\leq$ 0.003).

As with the light traffic results the individual results obtained by each algorithm under mixed traffic network environments can be seen in Figure 5.10.

It can be seen that when there are 285 UEs on the network the dSerFR dominates most of the baseline algorithms, that is it performs at least as well in both the 5th percentile throughput and average throughput metrics. The only exception to this is the maximum rate algorithm which optimises the average throughput at the expense of 5th percentile throughput. It can also be seen that none of the aSerFR results, while providing better absolute results, dominate the dSerFR results.

When there are 680 UEs on the network the dSerFR algorithm is outperformed by the SFR algorithm in terms of 5th percentile throughput. However the dSerFR does achieve better average throughput. It can also be seen that the dSerFR maintains a position on the Pareto front, often dominating a number of other solutions, including the baseline algorithms.

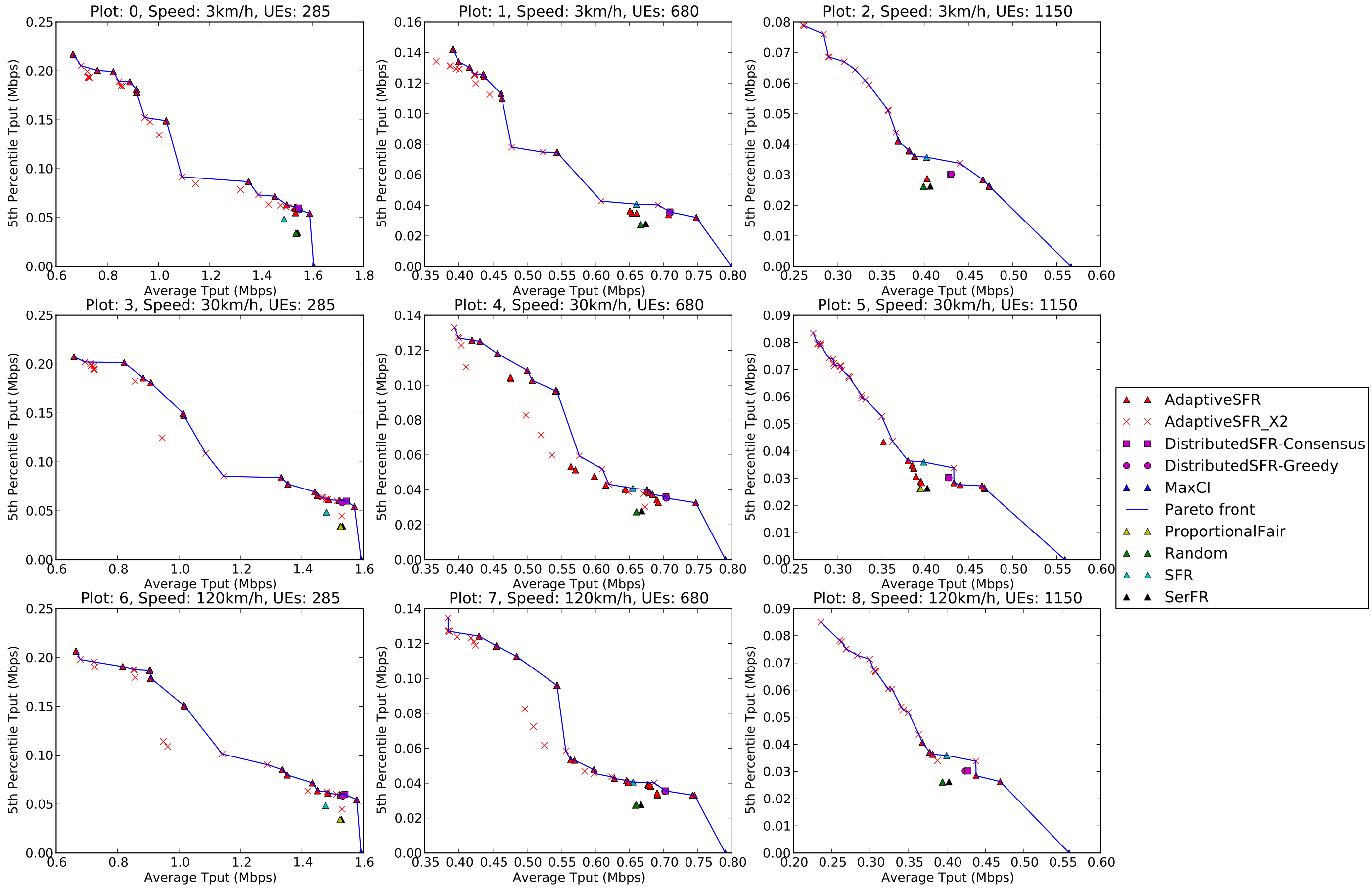


Figure 5.10: The 5th Percentile UE throughput vs average UE throughput in mixed traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

In the case where there are 1150 UEs on the network the SFR algorithm again achieves better 5th percentile throughput than the dSerFR algorithms. In these cases however it can be seen that the dSerFR results are dominated by those achieved by the aSerFR+X2 algorithm. In these cases the aSerFR+X2 algorithm achieves better 5th percentile throughput and average throughput than the dSerFR algorithm.

### 5.2.4 Heavy Traffic

The average UE throughput results can be seen in Figure 5.11. The average performance of each algorithm under heavy traffic networks appears very similar with the exception of the maximum rate algorithm which achieves a much better average UE throughput.

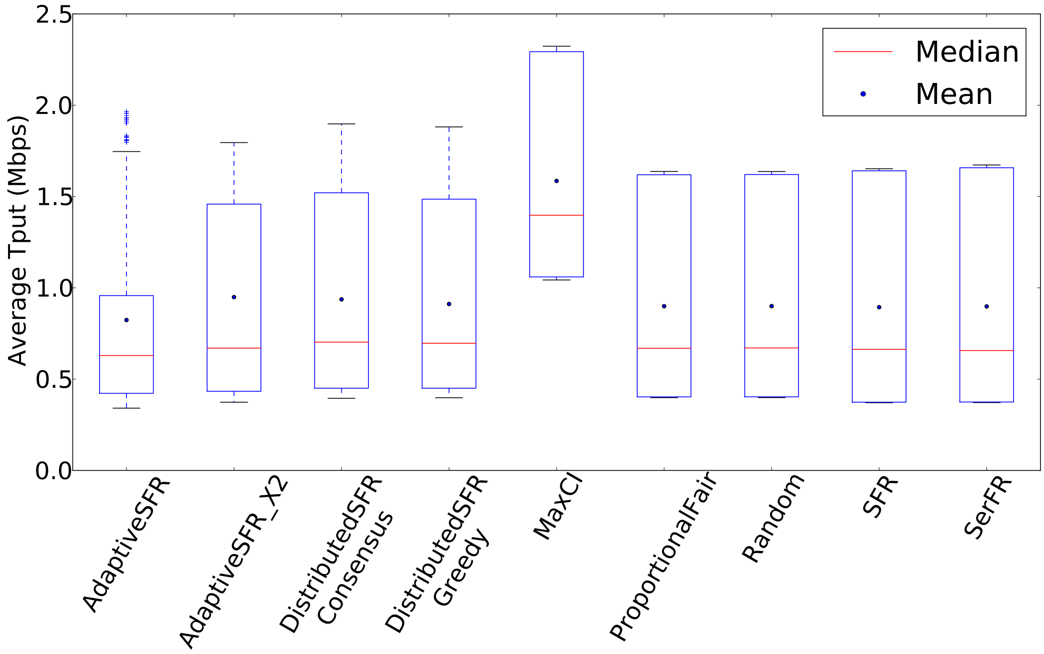


Figure 5.11: Boxplot showing the average UE throughput of each algorithm across all heavy traffic network environments.

The one-way ANOVA found a significant effect of algorithm ($F(8,1637) = 22.46$, $p = 0.001$). As suggested by Figure 5.11 the Bonferroni post-hoc shows that the only significant difference in performance is due the improved average UE throughput achieved by the maximum rate algorithm. The average mean difference between the maximum rate algorithm and all the other algorithms was 684kbps (all $p \leq 0.001$).

Figure 5.12 shows the results for each algorithm in terms of the 5th percentile UE throughput. It can be seen that there appears to be less variation in the results achieved by each algorithm than when under mixed traffic and light traffic loads. In particular it appears that the performance of the dSerFR algorithms achieved a similar level of performance to the aSerFR and aSerFR+X2 along with the baseline algorithms.

The one-way ANOVA again revealed a significant effect of algorithm ($F(8,1637) = 42.82$, $p \leq 0.001$). As suggested by Figure 5.12 the Bonferroni post-hoc comparisons showed that there was no significant difference between the results achieved by the dSerFR algorithms and any of the other algorithms, excepting the maximum

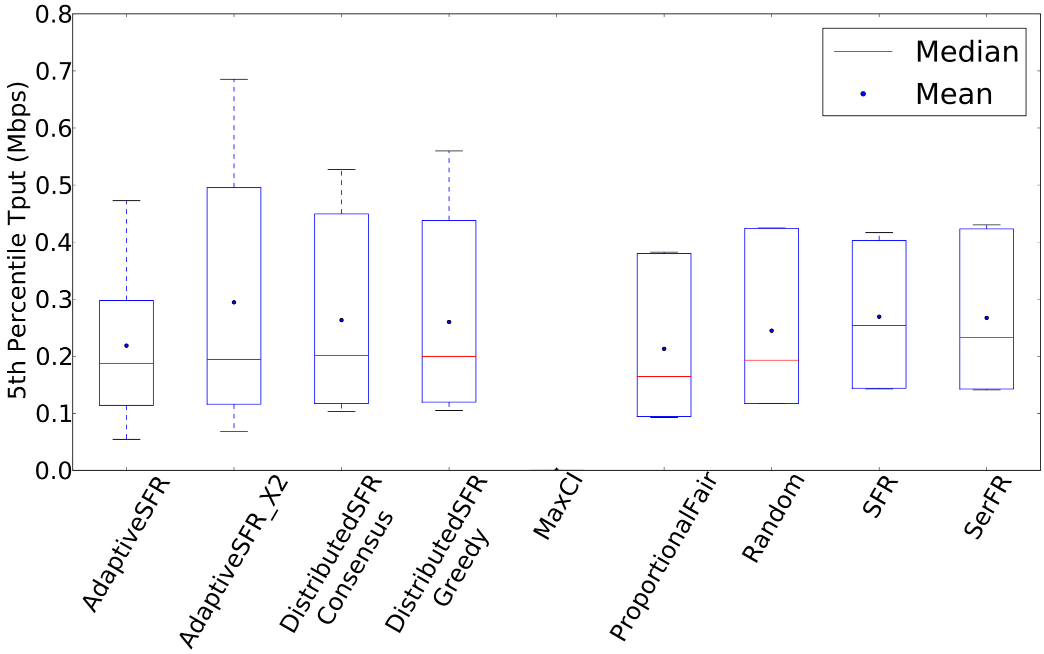


Figure 5.12: Boxplot showing the average 5th percentile UE throughput of each algorithm across all heavy traffic network environments.

rate algorithm which again performed worse than any of the others. In contrast to the mixed traffic environments the performance of the dSerFR algorithms was statistically indistinguishable from the aSerFR and aSerFR+X2 algorithms.

As can be seen from the individual results under heavy traffic in Figure 5.13 when 285 UEs are present on a heavily loaded network the dSerFR algorithm provides statistically significant improvements over the baseline algorithms in terms of 5th percentile throughput. For example in Plot 0, the dSerFR algorithms achieve a 5th percentile UE throughput of 469kbps, compared to the best performing of the baseline algorithms which was the random algorithm with 424kbps. This difference was found to be statistically significant ($t(18) = 2.2053$, $p = 0.0407$). While the baseline algorithms, in particular the SerFR algorithm, do appear to achieve better average UE throughput, this was not found to be statistically significant. For example in Plot 0 the SerFR algorithm achieved an average UE throughput of 1.672Mbps compared to the 1.596Mbps achieved by the dSerFR algorithms. However this was discovered to not be statistically significant ($t(18) = 1.2873$, $p = 0.2143$).

When 680 UEs are on the network both SFR and SerFR algorithms achieve the best absolute 5th percentile UE throughput, the dSerFR algorithms are able to

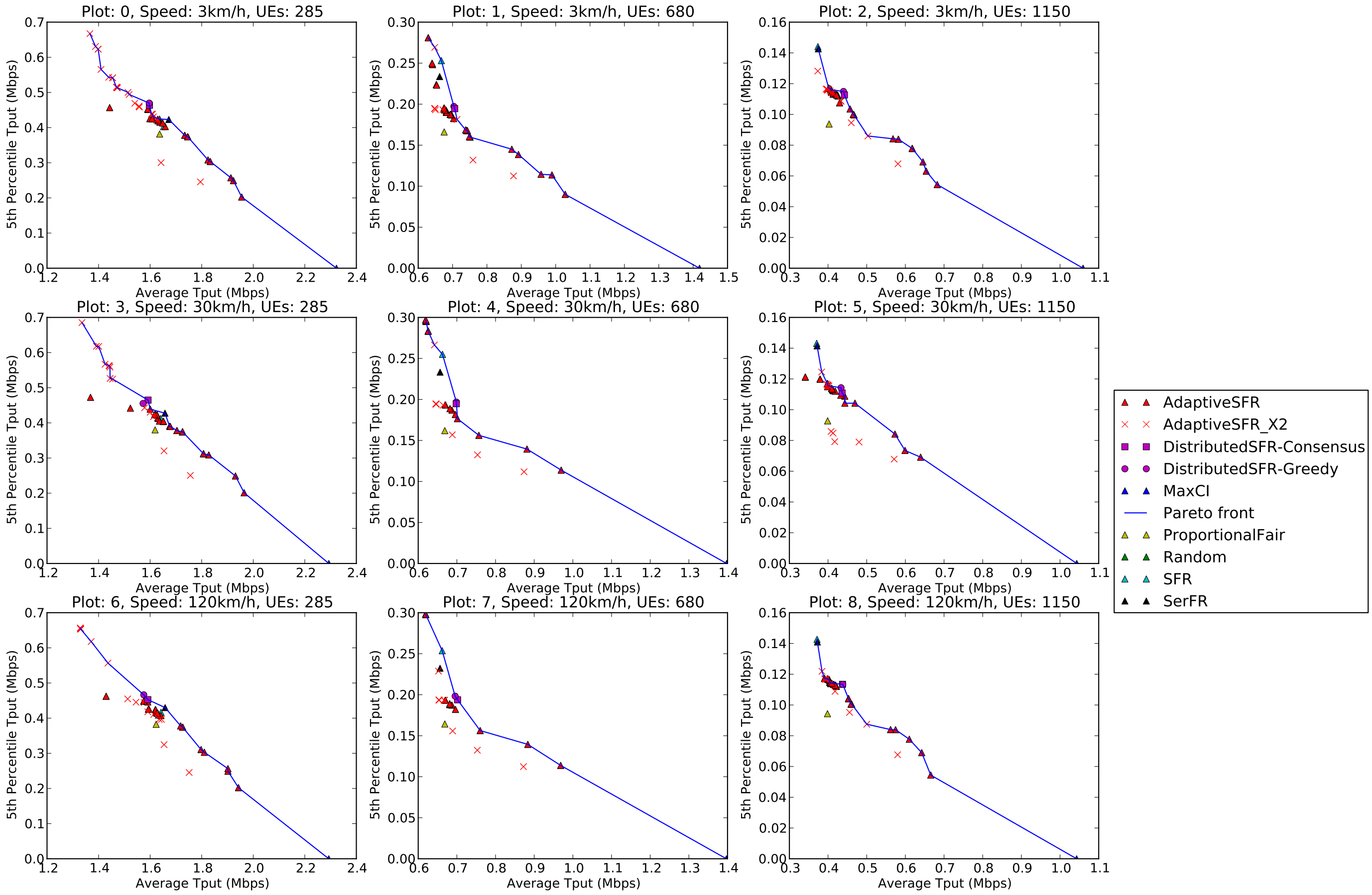


Figure 5.13: The 5th Percentile UE throughput vs average UE throughput in heavy traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

dominate some of the solutions discovered by the other algorithms, thus expanding the Pareto front.

In each network environment the dSerFR also appears on the Pareto front often dominating a number of solutions produced by the aSerFR, aSerFR+X2, SFR and SerFR algorithms.

### 5.2.5 Full Traffic

The average UE throughput results under Full traffic can be seen in Figure 5.14. The average performance of each algorithm under full traffic networks appears very similar with the exception of the maximum rate algorithm which achieves a much better average UE throughput.

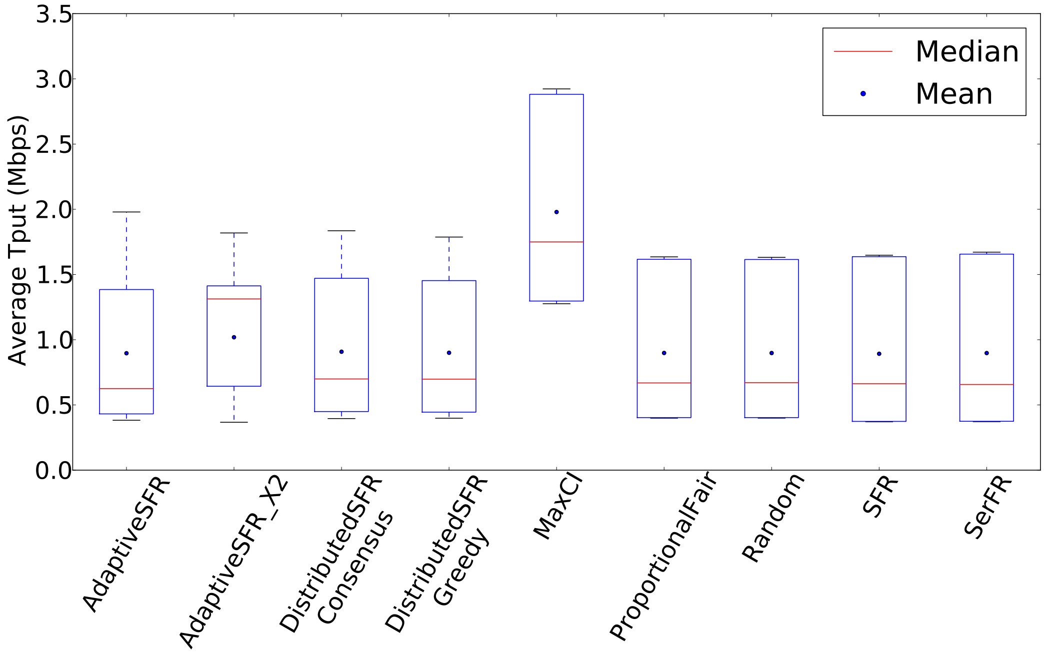


Figure 5.14: Boxplot showing the average UE throughput of each algorithm across all full traffic network environments.

The one-way ANOVA looking at the effect of algorithm on average UE throughput within full traffic network environments revealed a significant effect of algorithm (F(8,1768) = 47.32, p = 0.001). As suggested by Figure 5.11 the Bonferroni post-hoc shows that the only significant difference in performance is due the improve average UE throughput achieved by the maximum rate algorithm. The average mean difference between the maximum rate algorithm and all the other algorithms was 1.06Mbps (all $p \leq 0.001$).

Figure 5.15 shows each algorithms performance in terms of the 5th percentile UE throughput. It can be seen that the aSerFR+X2 algorithm appears to perform better than the others algorithms amount which there seems to be very little variation in results achieved.

The ANOVA confirmed a significant effect of algorithm (F(8,1768) = 50.41, $p \leq 0.001$). As suggested by Figure 5.15 the Bonferroni post-hoc comparisons showed that the dSerFR algorithms, both greedy and consensus based, performed significantly worse than the aSerFR+X2 with mean differences of 104kbps ($p \leq 0.001$) and 106kbps ($p \leq 0.001$) respectively. They also performed significantly better than

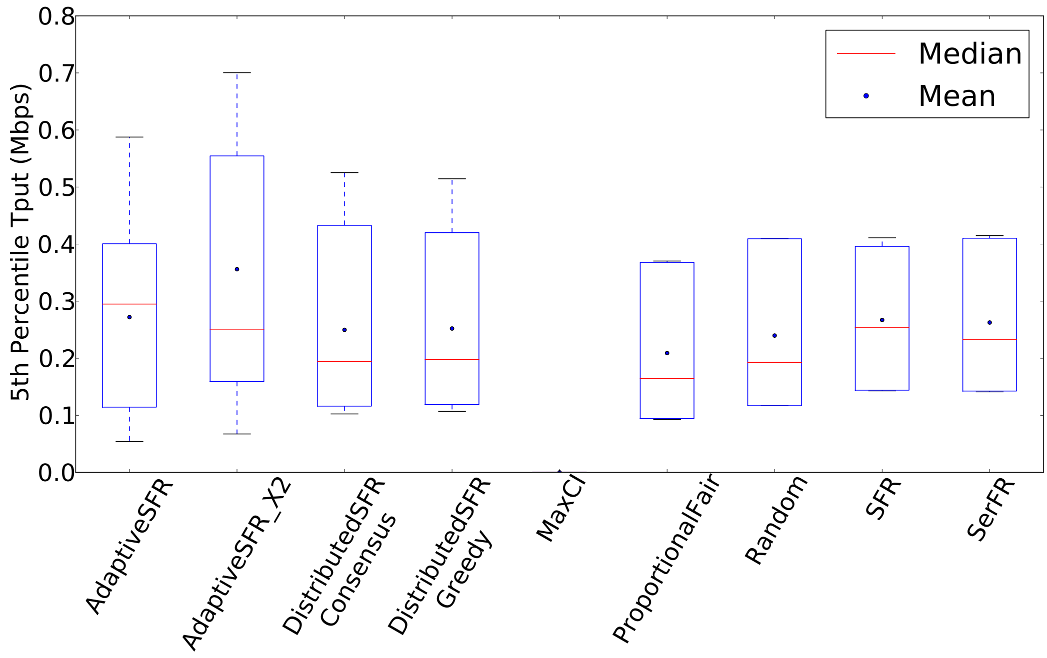


Figure 5.15: Boxplot showing the average 5th percentile UE throughput of each algorithm across all full traffic network environments.

the maximum rate algorithm which itself performed significantly worse than all the other algorithms (average mean difference was 263kbps, all $p \leq 0.001$).

The results achieved by each algorithm when the network is under full load can be seen in Figure 5.16. It can be seen that when there are only 285 UEs on the network the dSerFR achieves better 5th percentile throughput that the baseline algorithms. While the baseline algorithms do appear to achieve better average throughput the improvement is not statistically significant.

When there are either 680 UEs or 1150 UEs on the network both SFR and SerFR algorithms achieve statistically improved 5th percentile throughput when compared to the dSerFR algorithm. However, the dSerFR algorithm does achieve statistically improved average throughput when compared with the baseline algorithms.

It can also be seen that the results obtained from the dSerFR algorithm contribute in each case to the Pareto front defined by those results which are not dominated by any other.

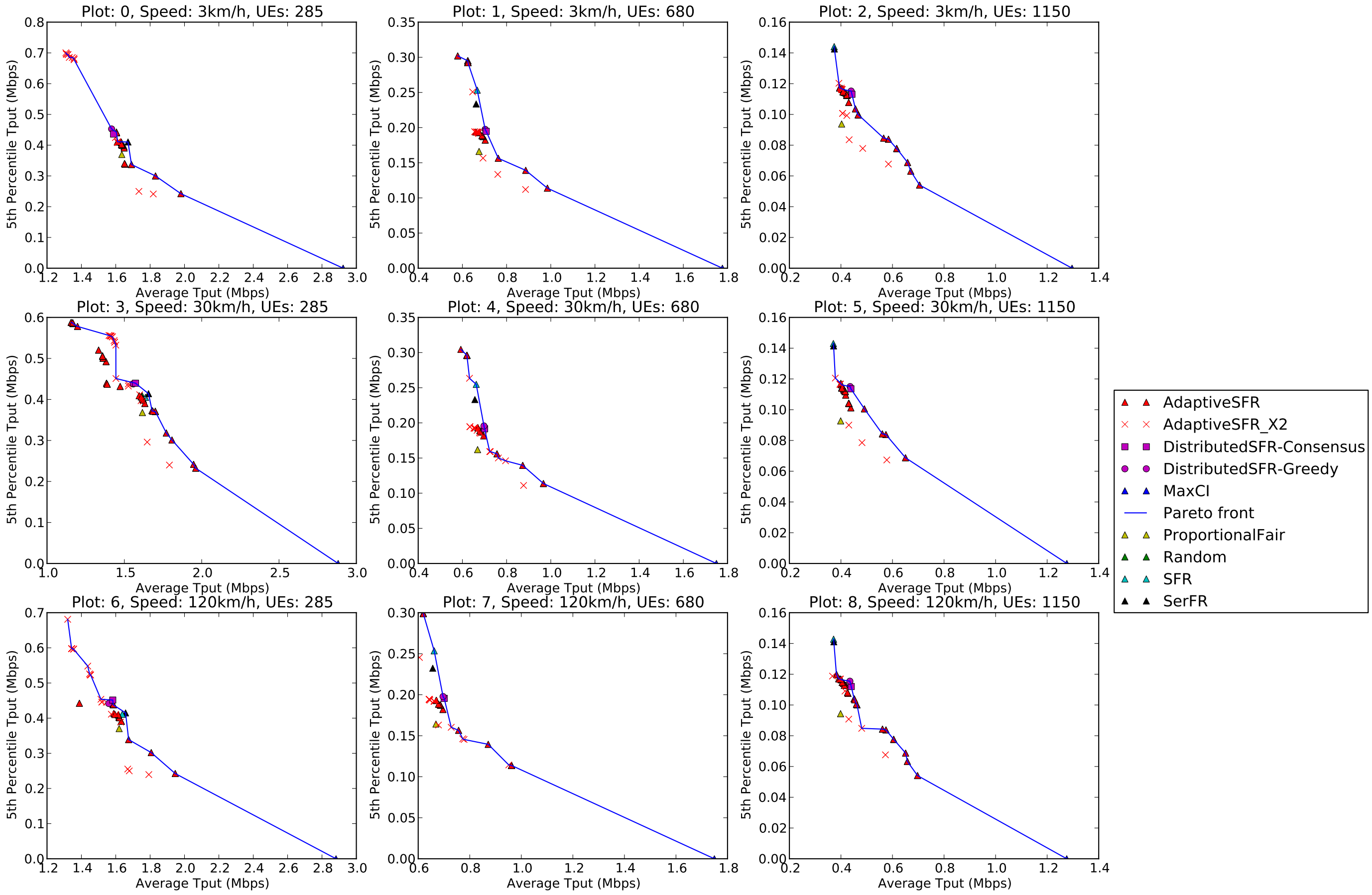


Figure 5.16: The 5th Percentile UE throughput vs average UE throughput in full traffic across nine network environments. Across the nine sub-plots the number of UEs increases from left to right while the UE speed increases from top to bottom.

### 5.2.6 Summary

The results reported above demonstrate that the dSerFR algorithms were, in every case, able to match the performance of the existing baseline algorithms in terms of 5th percentile UE throughput. In most cases the dSerFR algorithms were not able to match the performance of the aSerFR+X2 algorithm in terms of 5th percentile throughput but in general did achieve a similar level of average UE throughput.

## 5.3 Discussion

This chapter introduced a new ICIC scheme built on the aSerFR scheme presented in Chapter 3 and incorporated the X2 link to facilitate communication between the network nodes. This scheme was presented as an example of de-centralised organisation, that is allowing the eNBs themselves to run the optimisation in a distributed manner using the X2 link rather than requiring a centralised process. An experiment was described which compared the performance of this dSerFR scheme with the aSerFR and other schemes in order to see if this form of de-centralised organisation offered any performance improvements.

It can be seen in the results described in the preceding section that the dSerFR algorithm was not able to match the performance of the aSerFR or aSerFR+X2 algorithms in terms of 5th percentile UE throughput, though it was able to match the best performing of the baseline algorithms, the SFR and SerFR.

The fact that the performance of the dSerFR algorithms was not able to match that of theaSerFR or aSerFR+X2 algorithms may be because the only means of mutation in the dSerFR algorithm was the single point mutation operator which mutates a single gene in the chromosome. This means that the rate of mutation of the genes of the chromosomes, and consequently the exploration of the search space, will be slow. In contrast the Multi-Objective Genetic Algorithm (MO-GA) used to optimise the aSerFR and aSerFR+X2 algorithms also made use of a crossover operator which was able to introduce a greater level of genetic diversity into the chromosomes by combining two chromosomes to produce two distinct children. This process allows for a greater rate of genetic change and thus a quicker traversal of the search space. This lower rate of genetic change, due to the lack of crossover operation, could mean that the dSerFR algorithm took longer to explore the search space and therefore did not arrive at as good a solution as the aSerFR and aSerFR algorithms. This is also suggested by the fact that the performance of the dSerFR algorithm continued to improve over the duration of the simulation. This implies that the algorithm is still exploring the search space and has not arrived at a final optimal solution for each eNB.

It can also be noted from the above results how similar the results were between the greedy and consensus based dSerFR. This similarity suggests that both of these approaches to determining the fitness of a set of chromosomes do equally as well. It is possible that when taking the greedy approach, which attempts to optimise the 5th percentile throughput of the eNB without regard to its neighbours' performance, the results are generally good for the neighbouring eNBs also. This implies that an eNB optimising its own 5th percentile throughput may have a positive effect on its neighbours even without considering the mutation's effect on them.

Despite not improving the absolute 5th percentile throughput the dSerFR was able in many cases to extend the Pareto front by dominating some of the solutions created by the aSerFR and aSerFR+X2 schemes. This shows that the dSerFR scheme presented here was able to improve, in some areas of the search space, on those results generated by the other algorithms.

# Chapter 6

# Conclusion

The research described in this thesis was designed to investigate a number of approaches to using Self-Organising/Optimizing Network (SON) to enable network operators to better manage their networks. The particular focus of this work has been on the Inter-Cell Interference Coordination (ICIC) problem and using SON to optimise the scheduling of transmissions in such a way as to reduce the level of interference experienced by the network and thus automatically improve the performance of the network.

## 6.1 Main Findings

### 6.1.1 Local Organisation

The initial stage of the investigation attempted to observe the performance of some existing approaches to ICIC optimisation while also investigating whether some basic local optimisation could be used to improve the performance of the network.

A simulator was developed in order to investigate the performance of the existing ICIC schemes. An new ICIC scheme, Adaptive Softer Frequency Reuse (aSerFR), was developed based on the existing Softer Frequency Reuse (SerFR) scheme described in the literature. A number of additional parameters were added to the SerFR scheme along with a more dynamic approach to the allocation of the frequency bands. These parameters were then optimised using a Multi-Objective Genetic Algorithm (MO-GA) which focused on optimising the parameters for improved 5th percentile throughput and average throughput.

The first experiment, which tested the aSerFR algorithm across 36 network environments showed that in general the developed aSerFR scheme was able to improve the 5th percentile throughput achieved by the network by significant amounts. Indeed typically the aSerFR scheme produced the results with the best 5th percentile throughput, in particular when run on network environments which were not heavily loaded. Under the more heavily loaded network environments the aSerFR scheme generally matched the performance of the Soft Frequency Reuse (SFR) and SerFR schemes.

The Pareto front produced by the results from all the algorithms was greatly expanded by the addition of the aSerFR results. Due to the fact that the aSerFR scheme was optimised using a MO-GA the trade-offs between 5th percentile throughput and average throughput were made explicit. This would allow a network operator who wanted to use the aSerFR scheme to run the optimisation on their network and choose the set of parameters which best suited their requirements.

This shows that under some situations significant improvements in network performance can be obtained by using a local organisation strategy, such as an MO-GA, to tailor the configuration of the ICIC to the specific network environment.

### 6.1.2 Networked Organisation

The next stage of the investigation, presented in Chapter 4, was to determine whether a more network aware approach to organisation using the X2 interface to enable communication between the Evolved NodeBs (eNBs) could provide additional improvements to the performance of Long Term Evolution (LTE) networks.

A new ICIC scheme, Adaptive Softer Frequency Reuse + X2 (aSerFR+X2), was introduced which made use of the X2 link to facilitate direct communication between the eNBs on the network. In particular the eNBs were able to share performance information regarding the level of interference currently being experienced on particular Resource Blocks (RBs). When an eNB was under heavy load it could communicate to its neighbours a list of those RBs which were experiencing the most interference. This could then be used by the neighbouring eNBs to inform their scheduling decisions with the aim of reducing the interference experienced.

The results showed that in general the aSerFR+X2 scheme was able to further improve upon the gains made by the aSerFR scheme. In a number of cases the results produced achieved the best 5th percentile throughput when compared to all the other algorithms and in general the average 5th percentile throughput achieved in each traffic level either matched or exceeded that achieved by the other algorithms. The improvements were generally most pronounced when the network was under lighter loads rather than the heavier loads. It was also observed that in every case the results produced by the aSerFR+X2 scheme contributed to the Pareto front which contained the most fit results. Even in those cases where the 5th percentile throughput of the aSerFR+X2 scheme was only able to match the performance of the existing schemes, showing no showing no statistically significant improvement, there were often a number of solutions which did advance the Pareto front and dominate a number of the solutions produced by the existing algorithms.

The use of the X2 link in the aSerFR+X2 scheme and the results obtained demonstrate that significant improvements can be gained in terms of network performance by using a more network aware approach to self-organisation. In general the results obtained using this approach achieved better performance than those produced with only the local organisation embodied in aSerFR scheme.

### 6.1.3 De-centralised Organisation

The final stage of the investigation, described in this Chapter 5, considered the case of a more de-centralised approach to self-organisation. All of the previous approaches discussed could be considered centralised systems, where an optimisation process takes place on a central network node with the results pushed down to the actual nodes involved in running the Radio Access Network (RAN).

A simple distributed Genetic Algorithm (GA), De-centralised Softer Frequency Reuse (dSerFR), was developed which was designed to run on the eNBs themselves as opposed to some central network node. Each eNB was responsible for evolving its own set of configuration parameters in coordination with those eNBs in its immediate neighbourhood. This allowed each eNB to maintain a set of parameters which more accurately reflected the network environment in that specific location independently of its neighbours. The use of the X2 link was crucial in facilitating the organisation of this approach so that each eNB was aware of when it should begin evolving its configuration so as to avoid conflicts with its neighbours.

The results obtained from the final experiment described in Chapter 5 demonstrate that the dSerFR scheme was able to match the performance of the baseline algorithms but was not able to offer further improvements on the aSerFR and aSerFR+X2 algorithms in terms of 5th percentile. It was observed however that the majority of results produced by the dSerFR scheme did contribute to the Pareto front formed by the fittest of all the results taken together. This shows that the dSerFR scheme was able to successfully explore areas of the search space where the other algorithms had not. It was also observed that the performance of the dSerFR scheme continued to improve as the network ran. Since the optimisation progresses over time, in contrast to the more centralised schemes presented, this should mean that if run over longer periods of time the dSerFR algorithm could produce even better results.

This demonstrates that a more de-centralised approach to ICIC can provide additional options to network operators looking to balance the 5th percentile throughput of their network with the average throughput. The other main benefit of this type of de-centralised approach in general is that it is more autonomous that the more centralised approaches. In the centralised approach the network operator must run the optimisation process periodically on a network node and push the resulting parameters out to the eNBs on the network. In comparison, a de-centralised approach can run continuously on the eNBs themselves with essentially no external input required.

## 6.2 Future Work

There are a number of areas in this research which would benefit from further investigation. These additions can be classified in three main ways: improvements to the network simulator, further investigation and development into the ICIC schemes themselves and further investigation into alternative optimisation techniques.

### 6.2.1 Improvements to the Network Simulator

There are a number of features which could be added to the network simulator to build a more realistic picture of the ICIC schemes evaluated in this research.

#### 6.2.1.1 Improved Traffic Model

The simulator's traffic model is used to determine when and how much data arrives for a User Equipment (UE) during the course of the simulation. A UE with a light traffic level will only request a small amount of data from the eNB occasionally, whereas a UE operating with a full traffic level will continuously be requesting the maximum amount of data possible from the eNB. The current traffic model is probabilistic: each traffic level has a certain probability of requesting data and the amount of data requested is determined probabilistically. These specific probabilities are discussed in section 3.2.4. The traffic model used by the simulator in this research could model four traffic levels. This allowed the simulator to evaluate the ICIC schemes on a number of different network environments in order to build a more complete picture of their relative performance than that found in the literature.

However further improvements could be made to the traffic model used. For example the traffic model does not distinguish between different types of traffic. In reality there are a number of different types of traffic use that can be observed on a live network. 3rd Generation Partnership Project (3GPP) describes the following distinct traffic classes: conversational class, streaming class, interactive class, and background class [8]. Each of these traffic classes has different behaviour both in terms of the amount of data they require and also the sensitivity of that data to delay. For example the conversational traffic class, used to model the transmission of voice calls, has very stringent latency requirements while not necessarily requiring a large amount of data to be transmitted. In comparison the background traffic class which is used to model the delivery of email or the downloading of files, is much more insensitive to the latency but may request a much larger volume of data over time.

A number of concrete models have been proposed in the literature which model the data and latency requirements of these types of traffic classes. Implementing these in the network simulator would provide a more realistic view of the performance of the various ICIC schemes. For example [68] a number of models are presented that could be used to model Voice Over Internet Protocol (VoIP), that is traditional voice calls, video streaming, on-line gaming and HTTP based internet browsing.

These were not implemented in the simulator as part of this research since most of the existing studies are not concerned with the performance of algorithms on different traffic levels. The equivalent of the full traffic level used in this research, often called full buffer traffic mode, is generally used to assess the performance of ICIC algorithms in the literature [26, 22, 6, 103]. However, it has been found that the choice of traffic model, whether full buffered or not, can have a significant impact on the type of improvements that can be expected from ICIC schemes [51] and so further investigations with more realistic traffic models should enable a better understanding of the performance improvements offered by ICIC schemes under different network environments.

#### 6.2.1.2 Changes in Network Environment

Another possible improvement to the network simulator is the addition of more dynamic network environments. The network environments currently modelled by the network simulator can be thought of as static environments. The traffic level and the number of UEs on the network are fixed at the start of the simulation and do not change. This is only an approximation of a realistic network where UEs enter and leave the network at various times and where a single UE can change traffic level by making a new voice call or starting a new internet browsing session. In particular there are certain areas of operator networks, such as those serving heavy commuting routes, which typically experience significant changes in the number of UEs and traffic demand during the course of a day. Implementing such a dynamic network environment would allow the simulator to more accurately represent an LTE network. This would also allow the ICIC schemes to be evaluated on how well they performed under changing traffic loads as this is an area of research where further study would be helpful.

### 6.2.2 Further Investigation into ICIC Schemes

The investigations into improved ICIC described in this thesis has yielded significantly improved results in many cases. However there remain further areas of investigation which may enable further performance increases.

It was seen from the results described in the preceding chapters that the area where least improvements were observed was under those network environments which were heavily loaded either in terms of the number of UEs on the network or the traffic level of those UEs. It has been suggested in the literature [35] that under these heavily loaded network environments Inter-Cell Interference Randomisation (ICIR) may provide improved performance over ICIC. While ICIC was the focus of this research and as such ICIR has not been discussed at length or investigated, if it promised to improve upon the results delivered by ICIC then some further investigation would indeed be warranted.

There is also scope for further investigation of the performance of the ICIC presented in this thesis. In particular the dSerFR scheme described in Chapter 5 could potentially be improved in a number of ways. Due to time constraints the effect of a number of parameters of the algorithm have not been explored. For example the choice of selection algorithm used to determine which eNB should mutate could be explored to determine whether this has any significant effect on the algorithm's performance. In addition parameters like the mutation window could be further explored: what affect do different lengths of mutation window have on the results achieved by the dSerFR scheme?

### 6.2.3 Alternative Optimisation Techniques

As described in the previous chapters the MO-GA developed as part of this thesis provided the optimisation approach used to develop improved ICIC schemes. Two further promising related approaches could be investigated in this area to build on this research.

#### 6.2.3.1 Cellular GAs

Cellular GAs are are GAs in which the population takes the form of a grid with each vertex of the grid being occupied with a single candidate solution. During each iteration of the GA a new population is produced by applying the recombination and mutation operators to the existing population. The recombination and mutation operators do not apply to the whole population as in a standard GA. Rather when creating a new child solution for a given vertex, only those candidates in the immediate neighbourhood can be selected as parents. This leads to a sense of locality within the population know as isolation by distance as similar individuals tend to cluster into sub-populations in particular parts of the grid.

Multi-objective cellular GAs have also been developed [79] where the non-dominated candidate solutions are retained in a Pareto front. Experimental results suggest that when compared to standard MO-GAs the Pareto front obtained from a cellular MO-GA is significantly more diverse providing a more diverse range of Pareto optimal solutions.

Considering the spatial nature of the cellular MO-GA in light of the spatial nature of cellular networks such as LTE, using a cellular MO-GA may present an advantage. Allowing each eNB to represent a candidate solution at one point on the grid and allowing it to recombine with those eNBs in its neighbourhood could allow for a more diverse Pareto front to develop and potentially allow for areas of the network with different traffic patterns to develop solutions tailored to that area.

#### 6.2.3.2 Embodied Evolution

Embodied evolution [96] is a distributed, asynchronous GAs which was initially developed to solve three main problems in the field of evolutionary robotics by allowing the optimisation algorithm to run on a population of physical robots. Firstly, previous approaches did not scale well to many robots due many interactions between them. Secondly a large number of candidate solutions needed to be evaluated but previous methods of evaluation were too slow to evaluate these serially on a real robot. Yet thirdly, the evaluations needed to be done on a real robot since the low fidelity of simulations led to transference problems where simulation results were not repeatable in reality.

The embodied evolution scheme functions in the following way. Each robot attempts to complete a specific task and measure its own performance on the task. Prior to carrying out the task the control software for each robot is initialised with a random candidate solution which governs its behaviour and thus determines its fitness. As the robots go about their tasks they probabilistically broadcast a randomly chosen and mutated gene from their own chromosome. The rate of a robot's broadcast is proportional to its fitness with fitter robots broadcasting more often than their less fit counterparts. Robots receiving a broadcast accept it with a probability inversely proportional to their own fitness so that less fit robots are more likely to accept broadcasts from surrounding fitter robots.

These mechanisms allowed the embodied evolution approach to overcome the problems associated with previous approaches as follows. The population based approach allowed it to scale to the many robots required while also allowing for a large number of evaluations since these could be computed across a number of robots in parallel. The use of multiple real robots also avoided transference problems by running on real robots rather than on a simulation.

This work could be applied directly to the ICIC schemes developed in this thesis. An embodied scheme could be used to evolve the dSerFR ICIC scheme presented in chapter 5. In this case each of the eNBs would evaluate their own fitness and broadcast their randomly selected genes at a rate proportional to their own fitness. This approach could provide a low-bandwidth, low-complexity approach to developing a more distributed ICIC scheme.

## 6.3 Concluding Remarks

This thesis has described research into using self-organisation techniques as part of an ICIC approach to reduce interference and thus improve the performance of LTE networks. The original ICIC schemes presented in this work have been shown to significantly improve the performance of LTE networks under a large number of different network conditions. A number of further avenues for investigation have been described which could be used to further improve the performance of LTE networks going forward. It is hoped that the research presented in this thesis can not only inform industry regarding the current state of ICIC, but can serve as a foundation upon which further developments can be built.

# Bibliography

[1] 3GPP TR 25.913. Requirements for evolved UTRA (E-UTRA) and evolved UTRAN (E-UTRAN), December 2008.

[2] 3GPP TS 32.500. Telecommunication management; Self-Organising networks (SON): concepts and requirements, July 2008.

[3] 3GPP TS 36.104. TS 36.104: Base Station (BS) radio transmission and reception, May 2008.

[4] 3GPP TS 36.201. TS 36.201: LTE Physical Layer - General Description, November 2007.

[5] 3GPP. TR 36.814: Further advancements for E-UTRA physical layer aspects, 2010.

[6] 3GPP. TS 36.942: Radio Frequency (RF) system scenarios, 2010.

[7] 3GPP. TR 36.902: Self-configuring and self-optimizing network (SON) use cases and solutions, 2011.

[8] 3GPP. Quality of Service (QoS) concept and architecture, 2012.

[9] 3GPP. TS 25.996: Spatial channel model for Multiple Input Multiple Output (MIMO) simulations, September 2012.

[10] 3GPP. TS 36.214: Physical Layer - Measurements, December 2012.

[11] 3GPP. TS 36.420: X2 General Aspects and Principles, September 2012.

[12] 3GPP. TS 36.421: X2 Layer 1, December 2012.

[13] 3GPP. TS 36.422: X2 Signalling Transport, September 2012.

[14] 3GPP. TS 36.424: X2 Data Transport, September 2012.

[15] 3GPP. TS 36.211: Physical Channels and Modulation, March 2013.

[16] 3GPP. TS 36.212: Multiplexing and Channel Coding, March 2013.

[17] 3GPP. TS 36.213: Physical Layer Procedures, March 2013.

[18] 3GPP. TS 36.423: X2 Application Protocol (X2AP), March 2013.

[19] 3GPP TR 25.814. Physical layer aspects for evolved Universal Terrestrial Radio Access (UTRA), 2006.

[20] 4G++. 4G++: Advanced Performance Boosting Techniques in 4th Generation Wireless Systems: Inter-Cell Interference Coordination, 2012.

[21] Alcatel. Multi-cell simulation results for interference co-ordination in new OFDM DL. Technical Report R1-050694, 29th August 2nd September 2005, London, UK, September 2005.

[22] Alcatel. R1-060209: System simulation results for downlink interference coordination. Technical report, 3GPP, Helsinki, Finland, January 2006.

[23] NGMN Alliance. Informative list of SON use cases, April 2007.

[24] E. Amaldi, A. Capone, and F. Malucelli. Discrete models and algorithms for the capacitated location problems arising in umts network planning. In *Proceedings of the 5th international workshop on Discrete algorithms and methods for mobile computing and communications*, DIALM '01, pages 1–8, New York, NY, USA, 2001. ACM.

[25] Jeffrey G. Andrews, Arunabha Ghosh, and Rias Muhamed. *Fundamentals of WiMAX, Understanding Broadband Wireless Networking.* Prentice Hall, February 2007.

[26] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall. LTE: the evolution of mobile broadband. *Communications Magazine, IEEE*, 47(4):44–51, April 2009.

[27] G. Auer. On modeling cellular interference for multi-carrier based communication systems including a synchronization offset. In *WPMC03*, pages 290–294, 2003.

[28] T Bäck, D.B Fogel, and Z Michalewicz, editors. *Mutation Operators*. Institute of Physics Publishing, Bristol, 2000.

[29] T Bäck, D.B Fogel, and Z Michalewicz, editors. *Mutation Operators*. Institute of Physics Publishing, Bristol, 2000.

[30] Thomas Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996.

[31] Constantine A. Balanis. *Antenna Theory: Analysis and Design, 3rd Edition*. Wiley-Interscience, 3 edition, April 2005.

[32] David Beasley, David R. Bull, and Ralph R. Martin. An Overview of Genetic Algorithms: Part 2, Research Topics. *University Computing*, 15(4):170–181, 1993.

[33] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms: Part 1, fundamentals, 1993.

[34] Tobias Blickle and Lothar Thiele. A comparison of selection schemes used in genetic algorithms. Technical report, Gloriastrasse 35, CH-8092 Zurich: Swiss Federal Institute of Technology (ETH) Zurich, Computer Engineering and Communications Networks Lab (TIK, 1995.

[35] R. Bosisio and U. Spagnolini. Interference coordination vs. interference randomization in multicell 3gpp lte system. In *Wireless Communications and Networking Conference, 2008. WCNC 2008. IEEE*, pages 824–829, 2008.

[36] Patrice Calegari, Frederic Guidec, Pierre Kuonen, and Frank Nielsen. Combinatorial optimization algorithms for radio network planning. *Theoretical Computer Science*, 263(12):235 – 245, 2001.

[37] D. Chizhik, J. Ling, P.W. Wolniansky, R.A. Valenzuela, N. Costa, and K. Huber. Multiple-input-multiple-output measurements and modeling in manhattan. *Selected Areas in Communications, IEEE Journal on*, 21(3):321 – 331, apr 2003.

[38] Brian Classon, Ajit Nimbalker, Stefania Sesia, and Issam Toufik. *Channel Coding and Link Adaptation*, pages 207–241. John Wiley and Sons, Ltd, 2009.

[39] E. Dahlman, H. Ekstrom, A. Furuskar, Y. Jading, J. Karlsson, M. Lundevall, and S. Parkvall. The 3G Long-Term evolution - radio interface concepts and performance evaluation. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 1, pages 137–141, 2006.

[40] Erik Dahlman, Stefan Parkvall, Johan Skold, and Per Beming. *3G Evolution, Second Edition: HSPA and LTE for Mobile Broadband*. Academic Press, 2 edition, 2008.

[41] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast elitist multi-objective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6:182–197, 2000.

[42] Giovanna Di Marzo Serugendo, Marie-Pierre Gleizes, and Anthony Karageorgos. Self-organization in multi-agent systems. *Knowl. Eng. Rev.*, 20:165–189, June 2005.

[43] S.N. Donthi and N.B. Mehta. Joint performance analysis of channel quality indicator feedback schemes and frequency-domain scheduling for lte. *Vehicular Technology, IEEE Transactions on*, 60(7):3096–3109, 2011.

[44] M. Dorigo, V. Maniezzo, and A. Colorni. Ant system: optimization by a colony of cooperating agents. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 26(1):29 –41, feb 1996.

[45] Marco Dorigo. Ant colony optimization: A new meta-heuristic. In *Proceedings of the Congress on Evolutionary Computation*, pages 1470–1477. IEEE Press, 1999.

[46] Ericsson. Inter-cell interference handling for E-UTRA. Technical Report R1-050764, 3GPP, London, UK, September 2005.

[47] Ericsson. R1-050764: Inter-cell Interference Handling for E-UTRA. Technical report, Ericsson, London, UK, 2005.

[48] Ericsson. Downlink inter-cell interference co-ordination/avoidance evaluation of frequency reuse. Technical report, 3GPP, May 2006.

[49] Ericsson. Frequency hopping for E-UTRA uplink. Technical report, Ericsson, Seoul, Korea, 2006.

[50] Ericsson. On inter-cell interference coordination schemes without/with traffic load indication. Technical report, 3GPP, October 2006.

[51] Ericsson. R1-081093: Further considerations on the use of Inter-Cell interference indication, 2008.

[52] M.J. Feuerstein, K.L. Blackard, T.S. Rappaport, S.Y. Seidel, and H.H. Xia. Path loss, delay spread, and outage models as functions of antenna height for microcellular system design. *Vehicular Technology, IEEE Transactions on*, 43:487–498, 1994.

[53] I. Forkel, A. Kemper, R. Pabst, and R. Hermans. The effect of electrical and mechanical antenna down-tilting in UMTS networks. In *3G Mobile Communication Technologies, 2002. Third International Conference on (Conf. Publ. No. 489)*, pages 86–90, 2002.

[54] R. Ghaffar and R. Knopp. Interference suppression strategy for cell-edge users in the downlink. *Wireless Communications, IEEE Transactions on*, 11(1):154–165, 2012.

[55] Al Globus, Greg Hornby, Derek Linden, and Jason Lohn. Automated antenna design with evolutionary algorithms.

[56] I.A. Glover and P.M. Grant. *Digital communications*. Pearson Prentice Hall, 2010.

[57] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. MAddison Wesley, 1989.

[58] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989.

[59] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[60] David E. Goldberg and Kalyanmoy Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, 1991.

[61] Andrea Goldsmith. *Wireless Communications*. Cambridge University Press, New York, NY, USA, 2005.

[62] B. Hagerman, D. Imbeni, J. Barta, A. Pollard, R. Wohlmuth, and P. Cosimini. Wcdma 6-sector deployment - case study of a real installed umts-fdd network. In *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, volume 2, pages 703 –707, may 2006.

[63] John H. Holland. *Adaptation in natural and artificial systems*. MIT Press, Cambridge, MA, USA, 1992.

[64] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[65] J.M. Holtzman. Asymptotic analysis of proportional fair algorithm. In *Personal, Indoor and Mobile Radio Communications, 2001 12th IEEE International Symposium on*, volume 2, pages F–33–F–37 vol.2, 2001.

[66] T. Isotalo, J. Niemela, and J. Lempiainen. Electrical antenna downtilt in UMTS network. *Proc. 5th European Wireless Conference, Barcelona*, pages 265–271, 2004.

[67] P.D. Karamalis, N.D. Skentos, and A.G. Kanatas. Selecting array configurations for mimo systems: an evolutionary computation approach. *Wireless Communications, IEEE Transactions on*, 3(6):1994 – 1998, nov. 2004.

[68] Farooq Khan. *LTE for 4G Mobile Broadband*. Cambridge University Press, 2009.

[69] J. Knowles and D. Corne. The pareto archived evolution strategy: a new baseline algorithm for pareto multiobjective optimisation. In *Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on*, volume 1, pages –105 Vol. 1, 1999.

[70] Kyu In Lee, Kyung-Soo Woo, Yo Han Ko, Jae Young Ahn, and Yong-Soo Cho. An inter-cell interference cancellation method for ofdm cellular systems using a subcarrier-based virtual mimo. In *Vehicular Technology Conference, 2006. VTC-2006 Fall. 2006 IEEE 64th*, pages 1–5, 2006.

[71] Marco Mamei, Ronaldo Menezes, Robert Tolksdorf, and Franco Zambonelli. Case studies for self-organization in computer science. *J. Syst. Archit.*, 52:443–460, August 2006.

[72] C. Maple, Liang Guo, and Jie Zhang. Parallel genetic algorithms for third generation mobile network planning. In *Parallel Computing in Electrical Engineering, 2004. PARELEC 2004. International Conference on*, pages 229 – 236, sept. 2004.

[73] H. Meunier, E.-G. Talbi, and P. Reininger. A multiobjective genetic algorithm for radio network optimization. In *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, volume 1, pages 317–324 vol.1, 2000.

[74] Brad L. Miller and David E. Goldberg. Genetic algorithms, tournament selection, and the effects of noise. *Complex Systems*, 9:193–212, 1995.

[75] Ajay R. Mishra. *Fundamentals of Cellular Network Planning and Optimisation: 2G, 2.5G, 3G... Evolution to 4G.* John Wiley and Sons, 2004.

[76] T. Mitchell. *Machine Learning.* McGraw-Hill Higher Education, 1997.

[77] M. Moustafa, I. Habib, and M.N. Naghshineh. Efficient radio resource control in wireless networks. *Wireless Communications, IEEE Transactions on*, 3:2385–2395, 2004.

[78] M. Moustafa, I. Habib, and M.N. Naghshineh. Efficient radio resource control in wireless networks. *Wireless Communications, IEEE Transactions on*, 3:2385–2395, 2004.

[79] Antonio J. Nebro, Juan J. Durillo, Francisco Luna, Bernab Dorronsoro, and Enrique Alba. Mocell: A cellular genetic algorithm for multiobjective optimization. *International Journal of Intelligent Systems*, pages 25–36, 2007.

[80] J. Niemela and J. Lempiainen. Impact of mechanical antenna downtilt on performance of WCDMA cellular network. In *Vehicular Technology Conference, 2004. VTC 2004-Spring. 2004 IEEE 59th*, volume 4, pages 2091–2095 Vol.4, 2004.

[81] Byoung-Seong Park, Jong-Gwan Yook, and Han-Kyu Park. The determination of base station placement and transmit power in an inhomogeneous traffic distribution for radio network planning. In *Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th*, volume 4, pages 2051–2055 vol.4, 2002.

[82] Theodore Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition, 2001.

[83] Craig W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA, 1987. ACM.

[84] D.H. Ring. Mobile telephony - wide area coverage. Technical Memo TA4 47-1 60-37, Bell Laboratories, December 1947.

[85] RITT. Inter-cell Interference Mitigation based on IDMA. Technical report, RITT, Sophia Antipolis, France, June 2005.

[86] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach (2nd Edition)*. Prentice Hall, December 2002.

[87] Farag Sallabi and K. Shuaib. Modeling of downlink wireless fading channel for 3gpp lte cellular system. In *GCC Conference and Exhibition (GCC), 2011 IEEE*, pages 421–424, 2011.

[88] Siemens. Aspects of interference mitigation by coordination. Technical report, 3GPP, November 2005.

[89] Siemens. Interference mitigation considerations and results on frequency reuse. Technical report, 3GPP, September 2005.

[90] Siemens. R1-050738: Interference mitigation - Considerations and Results on Frequency Reuse. Technical report, 3GPP, London, UK, September 2005.

[91] Iana Siomina, Peter Varbrand, and Di Yuan. Automated optimization of service coverage and base station antenna configuration in UMTS networks. *Wireless Communications, IEEE [see also IEEE Personal Communications]*, 13:16–25, 2006.

[92] Fanglei Sun, Mingli You, Jin Liu, Zhenning Shi, Pingping Wen, and Jianguo Liu. Genetic algorithm based multiuser scheduling for single- and multi-cell

systems with successive interference cancellation. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*, pages 1230 –1235, sept. 2010.

[93] A. Triki and L. Nuaymi. Intercell interference coordination algorithms in ofdma wireless systems. In *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, pages 1 –6, may 2011.

[94] A.A. Triolo. Directions in cellular telephony and implications for power systems. In *Telecommunications Energy Conference, 2000. INTELEC. Twenty-second International*, pages 8–11, 2000.

[95] F.J. Villegas, T. Cwik, Y. Rahmat-Samii, and M. Manteghi. A parallel electromagnetic genetic-algorithm optimization (ego) application for patch antenna design. *Antennas and Propagation, IEEE Transactions on*, 52(9):2424 – 2435, sept. 2004.

[96] Richard A. Watson, Sevan G. Ficici, and Jordan B. Pollack. Embodied evolution: Distributing an evolutionary algorithm in a population of robots. *Robotics and Autonomous Systems*, 39:1–18, 2002.

[97] R.M. Whitaker, L. Raisanen, and S. Hurley. A model for conflict resolution between coverage and cost in cellular wireless networks. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, page 8 pp., 2004.

[98] Mehmet E. Aydin ; Raymond Kwan ; Wei Ding ; Joyce Wu. A genetic algorithm approach for multiuser scheduling on the lte downlink. In *Lecture Notes in Engineering and Computer Science*, volume 2198, 2012.

[99] Fan Xiangning, Chen Si, and Zhang Xiaodong. An Inter-Cell interference coordination technique based on users' ratio and Multi-Level frequency allocations. In *Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on*, pages 799–802, 2007.

[100] Xu Yang, Yapeng Wang, Dapeng Zhang, and Laurie Cuthbert. Resource allocation in lte ofdma systems using genetic algorithm and semi-smart antennas. In *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, pages 1 –6, april 2010.

[101] O.N.C. Yilmaz, S. Hamalainen, and J. Hamalainen. Analysis of antenna parameter optimization space for 3gpp lte. In *Vehicular Technology Conference Fall (VTC 2009-Fall), 2009 IEEE 70th*, pages 1 –5, sept. 2009.

[102] Jun Zhang and J.G. Andrews. Adaptive spatial intercell interference cancellation in multicell wireless networks. *Selected Areas in Communications, IEEE Journal on*, 28(9):1455–1468, 2010.

[103] Xunyong Zhang, Chen He, Lingge Jiang, and Jing Xu. Inter-cell interference coordination based on softer frequency reuse in OFDMA cellular systems. In *Neural Networks and Signal Processing, 2008 International Conference on*, pages 270–275, 2008.