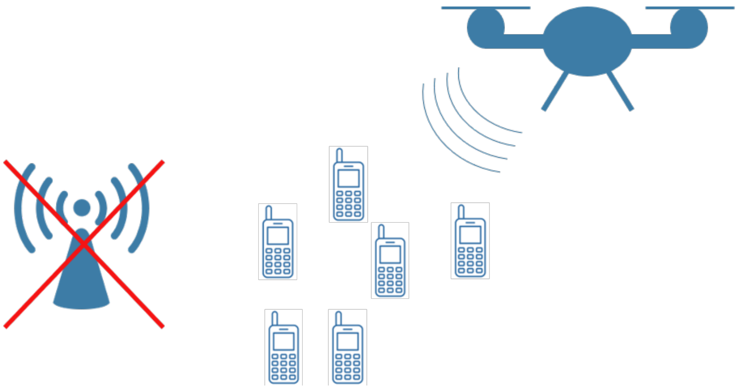


4G Network Patch using Drones

Diogo Santos Moura Gonçalves

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisor(s): Prof. António Manuel Raminhos Cordeiro Grilo

Examination Committee

Chairperson: Prof. Horácio Cláudio De Campos Neto
Supervisor: Porf. António Manuel Raminhos Cordeiro Grilo
Member of the Committee: Prof. Pedro Joaquim Amaro Sebastião

November 2018

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

Firstly, I would like to thank my supervisor, Prof.António Grilo, for all the support during the elaboration of this work. Thank you for allowing this work to be my own, but steering me in the right direction whenever it was needed.

Secondly, I would like to thank all of the friends who shared my despairs and successes, not only during this period, but also during my passing at IST.

Thirdly, I would like to direct some words to NEEC and JEEC. Being part of those groups really marked my course in the university. Not only because of all I learned here, but specially, because of the friendships that I made.

Lastly, I would like to thank my family and my most close friends for the unconditional love and support.

Resumo

O principal objetivo desta tese é estudar e propor uma nova alternativa ao patching de redes móveis. A solução implementada faz uso de drones para responder ao problema e foca-se no processo de escolha das melhores posições para os drones. Foi desenvolvido um algoritmo genético para superar este desafio que tira partido da distribuição não uniforme das necessidades de tráfego dos vários utilizadores da rede. Os drones tiram partido de estações base mais distantes para fornecer cobertura e capacidade numa determinada zona. Foi desenvolvido um simulador para avaliar a qualidade das posições retornadas pelo algoritmo. Esta provou ser uma maneira eficaz de calcular as posições dos drones. Foram utilizadas duas métricas para comparar o novo algoritmo contra soluções já existentes: o número de utilizadores servidos, e o débito binário total que é servido. Quando comparado com outras soluções, o novo algoritmo mostrou conseguir manter o número de utilizadores servidos sem grandes alterações, e melhorar até 24% o débito binário fornecido.

Palavras-chave: *Patch* de rede, UAV, LTE, Algoritmo Genético.

Abstract

The main goal of this thesis is to study and propose a new approach to network patching in mobile networks. The developed solution implements network patching making use of UAVs. The biggest challenge is to define the best positions to deploy the UAVs. A genetic algorithm was developed to tackle this problem. The developed algorithm takes advantage of the non-uniform distribution of data rate requirements among users to better position the UAVs. The UAVs make use of distant base stations to provide extra coverage or capacity in a given area. A simulator was developed to access the quality of the positions. The developed algorithm showed to be an effective way to compute the positions of the UAVs. Two metrics were analyzed to measure the effectiveness of the algorithm: the number of served users, and the total data rate served. Compared to another existing solutions, the algorithm was able to keep the number of served users in every scenario tested, and increase up to 24% the data rate provided.

Keywords: Network Patching, UAV, LTE, Genetic Algorithm.

Contents

List of Tables

List of Figures

Glossary

3GPP 3rd Generation Partnership Project. 6

ANACOM Autoridade Nacional de Comunicações. 6

BS Base Station. 11

CAPEX CAPital EXPenditure. 19

CDMA Code Division Multiple Access. 6

CoMP Coordinated Multipoint. 6

CQI Channel Quality indicator. 11

DL downlink. 6

DR Data Rate. 31

E-UTRAN Evolved UMTS Terrestrial Radio Access Network. 6

eICIC Enhanced Inter-Cell Interference coordination. 6

eNB evolved Node B. 7

EPC Evolved Packet Core. 6

FDD Frequency Division Duplex. 11

FEC Forward Error Correction. 19

GA Genetic Algorithm. 4

GSM Global System for Mobile Communications. 5

HSPA High Speed Packet Access. 6

HSS Home Subscriber Server. 8

IoT Internet of Things. 6

ITU International Mobile Telecommunications. 6

LOS Line of Sight. 22

LTE Long Term Evolution. 4

MAC Medium Access Control. 9

MIMO Multiple-Input Multiple-Output. 6

MME Mobility Management Entity. 8

NAS Non Access Stratum. 9

OFDM Orthogonal Frequency Division Multiplexing. 6

OFDMA Orthogonal Frequency-Division Multiple Access. 9

OPEX OPerational EXPenditure. 19

P-GW Packet Data Network Gateway. 8

PAPR Peak to Average Power Ratio. 10

PDCP Packet Data Convergence Protocol. 9

PHY Physical Layer. 9

RB Resource Blocks. 11

RBG Resource Block Groups. 14

RE Resource Element. 11

RF Radio Frequency. 11

RLC Radio Link Control. 9

RNs Relay Nodes. 18

RRC Radio Resource Control. 9

S-GW Serving Gateway. 8

SC-FDMA Single Carrier Frequency Division Multiple Access. 10

SDMA Space Division Multiple Access. 10

SNR Signal to Noise Ratio. 3

TDD Time Division Duplex. 11

TDMA Time-Division Multiple Access. 5

UAV Unmanned Aerial Vehicle. 3

UE User Equipment. 6

UICC Universal Integrated Circuit Card. 6

UL uplink. 6

UMTS Universal Mobile Telecommunications System. 6

USIM Universal Subscriber Identity Module. 6

VRBs Virtual Resource Blocks. 14

Chapter 1

Introduction

This chapter is an introduction to the work of this thesis. It will be shown why is important to study this subject and what are its implications on some of the application scenarios.

1.1 Technological Concept

If we consider last years evolution of the telecommunications systems, we must acknowledge that this technology has grown at a very fast pace and has had a massive adoption by the general population all around the globe. In 2017 there were already 7.8 billion mobile subscribers worldwide, and that number is expected to grow up to 9.1 billion by 2023 [1]. If we realize that the world population is under the 8 billion mark, these numbers become even more staggering.

As internet continues to evolve and grow, users are more than ever expecting to access it everywhere. Proof of that is the increasing demand for data traffic on mobile. As we can see from Figure 1.1, there is an exponential increase in the demand for mobile data traffic, in particular on the smartphones.

The demand for faster and better telecommunications systems does not come solely from the private market. Public security entities grow ever more dependent on communications to efficiently operate. In 2017, in Portugal, failures on SIRESP, the national emergency network operator, were pointed out as a big contributing factor for public disasters with fires. That year, the exposure of base stations and network connecting cables to fire lead to the breakdown of communications between public security entities. So it is important to develop either more resilient systems to these kind of situations, or better solutions for replacing the network elements that are more exposed.

As this is a very competitive market, telecommunications operators are permanently working for increasing their coverage, their capacity and for keeping the network with as close as possible to 100% availability. There are however a number of factors that can compromise these goals:

- ***Environmental Factors:*** Meteorologic/Natural events such as floods, earthquakes or fires that destroy or compromise some parts of the network. Usually these events cause a temporary lack of coverage in certain geographic places.

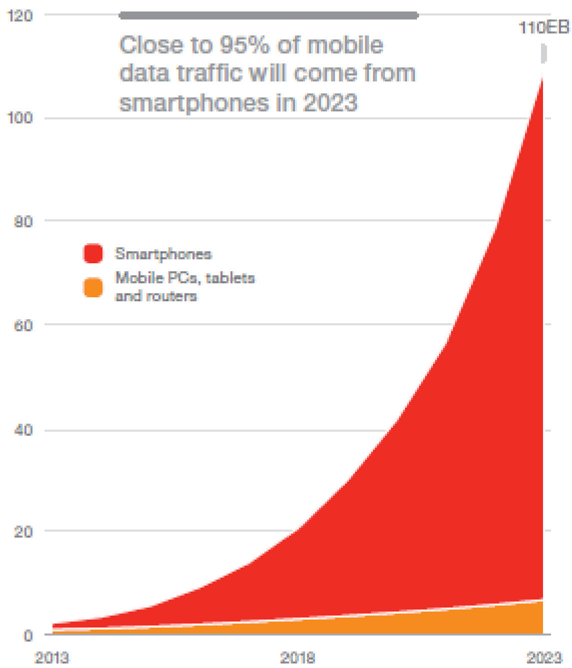


Figure 1.1: Global Mobile Traffic (ExaBytes per Month). Extracted from [1].

- ***Usage Peaks:*** Since network components are often expensive, engineers design the network for a certain amount of users in a specific area. However, sometimes there are special events that gather an unusual amount of people in a relative small space. That can cause lack of capacity in a cell.

- ***Components Breakdown:*** Some components of the network sometimes stop working properly causing disturbances on the telecommunications operation.

Some of these outages are predictable and can be planned ahead. In these cases operators develop dedicated projects to compensate for these expected failures. These operations are expensive and consume human resources, so it is important to keep investigating new and more efficient ways to solve these outages.

1.2 Solution Overview

It is possible to divide outages into two categories: expected and unexpected. The first is the case of a social event like a music festival or a sports competition, the other may be a flood or a fire. The compensation of an outage that is a result of such events is called a network patch. There are currently several ways to correct network outages, these vary in the amount of resources and time they require.

The simplest techniques involve tilting the existing antennas of base stations in order to optimize the coverage and capacity of a certain cell. Tilting can be mechanical, where the antenna physically changes its orientation; or electrical, where the antenna's radiation pattern is modified. This method is not very time or resource consuming, but is rather limited in its effect, since there is not an increment to the network resources in that area.

Another way to compensate for outages is to deploy more base stations to a certain area. This is a very time and resource consuming operation, it requires the engineers to know the geographical specifications, and to make good guesses on where the users will be. After all the planning is finished, base stations have to be moved to the site, which is also a rather expensive operation.

Overall, these approaches are non-adaptive over time, the planning takes place without knowing where the users will be and they are time and resource consuming. Taking this in consideration, due to their capability to freely move and deploy on-demand, drones are currently expected to help alleviate some of these problems in the future. In this work, the terms drone and Unmanned Aerial Vehicle (UAV) are used interchangeably.

The use of UAVs in telecommunication patches allows engineers to re-frame a problem that has always existed. With this new tool, it is possible to radically change the position of the antennas in real time. Consequently, if from one moment to another an important change in user positioning or traffic patterns takes place, we should use the maximum information possible about current user locations and traffic characteristics to relocate the UAVs. Currently, only the positioning of the users is used to chose the positions of UAVs, but since this process will be taking place in real-time, there are other factors that could help achieve a bigger network efficiency. One of those cases is the required data throughput that users are demanding at a particular moment.

Intuitively, if one user is requiring more resources, we should be able to approximate an UAV from her/him in order to improve its Signal to Noise Ratio (SNR), thus increasing the overall satisfaction of the users. It is important to keep in mind that the purpose of this work is not to make an extended study of the fairness of this approach. The driver of this work is to improve the signal of users that require a large amount of resources, specially in those situations where we do not need to compromise on the quality of service of users who need less resources.

The problem of finding the best position for the UAVs can be formulated in an optimization problem. Since there are several factors that we want to optimize, this work uses a genetic algorithm to tackle the problem.

1.3 Objectives

The main goal for this thesis is to analyze the current solutions for mobile networking patching and propose a solution that outperforms these methods. The proposed solution is to be tested and compared against some other solutions currently available. One specific goal is to analyze the effectiveness of a genetic algorithm to calculate the positions of the drones.

The new algorithm should not serve less users and should increase the data rate available for the individual users.

1.4 Contributions

This work implements a genetic algorithm for determining the position of UAVs. These UAVs will be providing mobile service for users.

The developed solution is compared against other algorithms for UAV placement. To achieve that, the algorithms are simulated, and the results compared.

1.5 Thesis Outline

This Thesis is divided into 5 chapters. The second chapter is meant to give some information on the main topics approached in this thesis. It begins by addressing the Long Term Evolution (LTE), explaining its main characteristics, architecture and services provided. Then, UAVs are addressed. There are some general considerations about them, a proposal of how we can classify them and a discussion of the roll they can have on this problem. After this, the Genetic Algorithm (GA) are presented. Their main components and their evolution processes are explained. In the end of this chapter, the K-Means Algorithm is presented. This algorithm was already proposed to tackle this problem and will serve as a baseline for the thesis.

Chapter 3 intends to explain the proposed solution. Firstly with the systems global architecture, and then focusing on their main components and explaining how they would be implemented. On the second half of this chapter, all the implementations made for this work are explored. The specific decisions relating the implementations are presented and the chapter ends with the description of the simulator developed to test the proposed solution.

Chapter 4 depicts the experiments performed. All the configurations of the simulator are showed and discussed. Then, a broader discussion on the results is made.

This thesis ends in the chapter 5 with a global discussion of the work. The main achievements are presented, and some future work on this subject is proposed.

Chapter 2

Background and Related Work

This chapter is going to address background and related work on three topics that concur for this thesis: LTE, UAVs and GAs. Firstly, LTE communications will be addressed, starting with a brief evolution of this technology, followed by a description of its main components. Then, the UAV universe will be addressed, from its evolution, to their current applications. It will be shown why UAVs are a promising tool for increasing the resilience of mobile telecommunications networks. Finally, GAs will be explored, explaining the main components and operators of these type of algorithms.

2.1 LTE Fundamentals

This section was mainly based on [2], [3], [4], [5] and [6]. It addresses the LTE, first with a brief evolution of this technology, then the architecture is explained, then the main mechanisms of the radio interface are explored. After that, the scheduling of radio resources is analyzed and before ending with the mechanisms of relaying, we describe the main factors that have implications in link capacity and the services that are provided in LTE's network.

2.1.1 Mobile Technology Evolution

To fully understand the LTE-Advanced network, it is useful to know some of the evolution of modern telecommunications networks. In 1979, the first commercial cellular network (1G) was launched in Japan. This analog technology was later implemented in several countries such as US, UK and Canada. Because this system was analog, it had a very low spectrum efficiency for today's standards, and was targeted almost exclusively for business users.

The Global System for Mobile Communications (GSM) was firstly deployed in 1987. This was the first digital cellular network, and became the standard for the 2nd generation (2G) of cellular systems. This system allowed the introduction of telecommunications as a consumer product for the first time. The shift to digital allowed the improvement of modulation, voice codecs and security. This second generation also implemented for the first time Time-Division Multiple Access (TDMA) to serve multiple subscribers. Qualcomm developed Interim Standard 95 (IS-95) and also brought Code Division Multiple Access (CDMA) to the 2G in North America and Korea.

In 1998 the 3rd Generation Partnership Project (3GPP) and 3GPP2 were founded to evolve CDMA and create the 3rd generation (3G) cellular standard. While 3GPP has worked standardization around GSM network, 3GPP2 continued to evolve the IS-95 network. This system was designed taking into consideration the user's demand for high data rates because of the use of Internet on the mobile phones. The data rate for this standard increased over time, going from the 384/128 kbps for downlink (DL) and uplink (UL) in the Universal Mobile Telecommunications System (UMTS) to the 28/11 Mbps in High Speed Packet Access (HSPA).

As the CDMA based network began to reach the limit for accomodating the demand for wireless data traffic, 3GPP decided to develop a standard based on a new access technology, the LTE. This standard adopted Orthogonal Frequency Division Multiplexing (OFDM) instead of CDMA for multiple access techonology, and used Multiple-Input Multiple-Output (MIMO) techonologies to increase even further the spectral efficiency.

In March 2008, 3GPP started to enhance the LTE in order to meet the International Mobile Telecommunications (ITU) requirements for the fourth generation (4G) evolution. This generation is called the LTE-Advanced and required at least the following techniques: carrier agregation, advanced MIMO, wireless relays, Enhanced Inter-Cell Interference coordination (eICIC), and Coordinated Multipoint (CoMP) transmission/reception. Initial LTE requirements demanded a peak data rate of 100/50 Mbps, however, this requirement was exceeded and the system is capable of providing 300/75 Mbps.

Currently there is a lot of research into the next generation, the fifth generation (5G). There is still a lot of uncertainty about the details of this new technology. However, 5G is expected to provide a significant increase in speed and a significant reduction in latency. It is also the first time Internet of Things (IoT) is being seriously considered in the standard definition. Service providers will have to build their networks in order to deal with the increased demands of network capacity distribution, traffic management and operations optimization. Many countries have already started to make way for this new technology, the Autoridade Nacional de Comunicações (ANACOM) for example, has already defined the road map for the release of specific bandwidths where 5G will operate [7].

The evolution of the number of subscribers by technology is showed in Figure 2.1. As one can see the first generations of mobile networks have virtually disappeared. Ericsson predicts that clients will continue to abandon subscriptions based on GSM alone and embrace LTE and 5G services.

2.1.2 LTE Architecture

Figure 2.2 shows the High Level Architecture of the LTE system, composed of three main components: the User Equipment (UE), the Evolved UMTS Terrestrial Radio Access Network (E-UTRAN) and the Evolved Packet Core (EPC).

The UE is any device where the data stream terminates. Each UE needs a Universal Integrated Circuit Card (UICC) that runs an application known as the Universal Subscriber Identity Module (USIM),

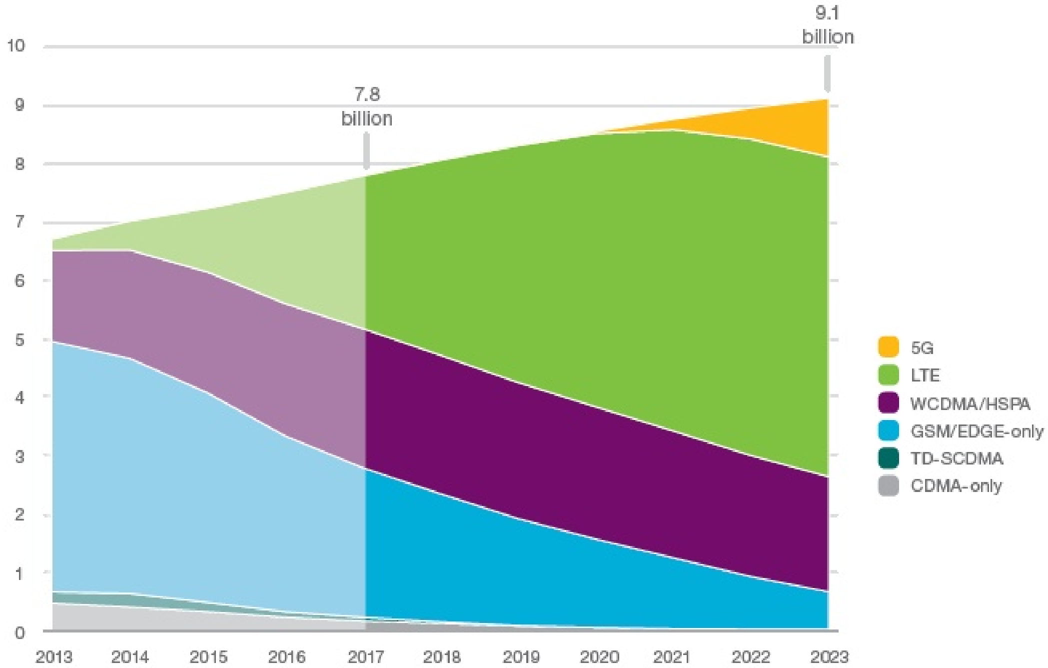


Figure 2.1: Mobile Subscribers By Technology (billions). Extracted from [1].

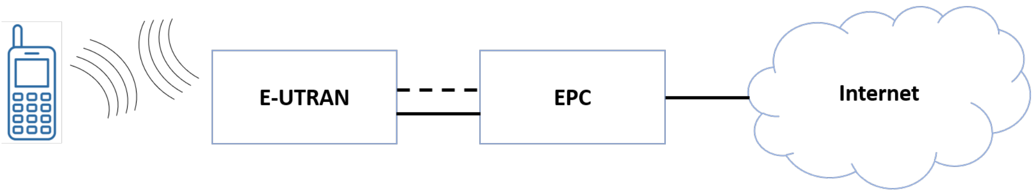


Figure 2.2: LTE architecture. Adapted from [3].

which stores user-specific data such as the user's phone number and home network identity. The USIM is also responsible for carrying out several security-related calculations, using secure keys that the smart card stores.

The Figure 2.3 shows the E-UTRAN architecture which handles the radio communications between the mobile and the evolved packet core. This element has just one component, the evolved Node B (eNB).

Each mobile is allocated to only one eNB at a time, and each eNB is responsible for controlling the mobiles in one or more cells. The eNB has two main functions. Firstly, the eNB sends radio transmissions to all its mobiles on the downlink and receives transmissions from them on the uplink, using analog and digital signal processing functions on the LTE air interface. Secondly, the eNB controls the low-level operations of all its mobiles, by sending them signalling messages, such as handover commands, that relate to those radio transmissions.

Each eNB has two interfaces with the rest of the network. One is used to communicate with the EPC. The other is not mandatory, and is used to communicate directly with other eNBs for signalling and packet forwarding during handover. The latter interface allows to bypass the EPC to minimize the flow of control data.

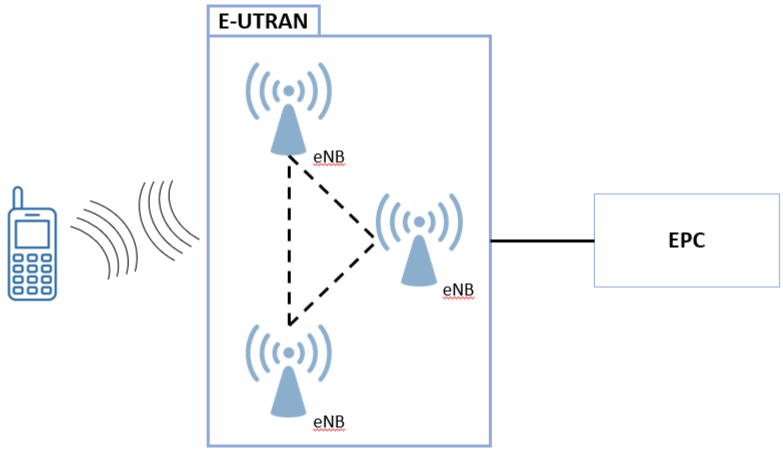


Figure 2.3: E-UTRAN architecture.

Figure 2.4 shows the main components of the EPC. The Home Subscriber Server (HSS) is a central database that contains information about all the network operator's subscribers.

The Packet Data Network Gateway (P-GW) is the EPC point of contact with the outside world. From here data packets are sent to the internet or the IP multimedia subsystem.

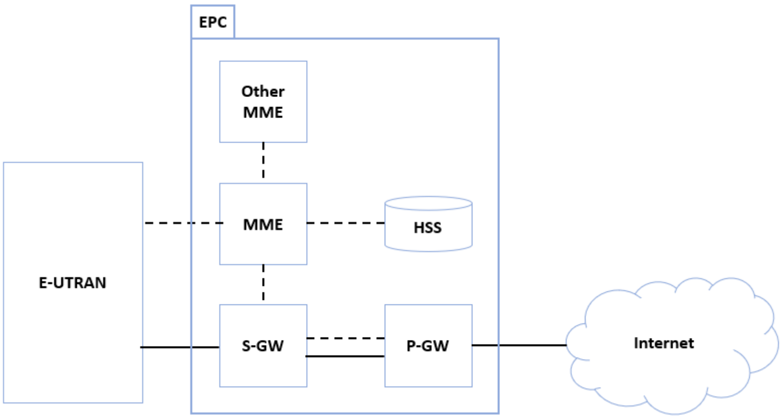


Figure 2.4: EPC architecture. Adapted from [3].

The Serving Gateway (S-GW) is responsible for routing and forwarding the data between the base station and the P-GW. Each S-GW serves all the mobiles of a certain geographic region, if a mobile changes geographic region it's reassign another S-GW.

The Mobility Management Entity (MME) controls the high-level operation of the mobile such as security and the management of data streams that are unrelated to radio communications. As the S-GW, the MME is responsible for a certain geographic region and mobiles are assign to MMEs according to their geographic coordinates.

2.1.3 LTE Protocol Stack

To better understand the functions each element in the LTE network performs it is important to know which protocols they implement. Figure 2.5 represents the protocols present in these elements. There is a division between the Non-Access Stratum and the Access Stratum. Furthermore, the protocols are divided to whether they belong to the control plane or the user plane.

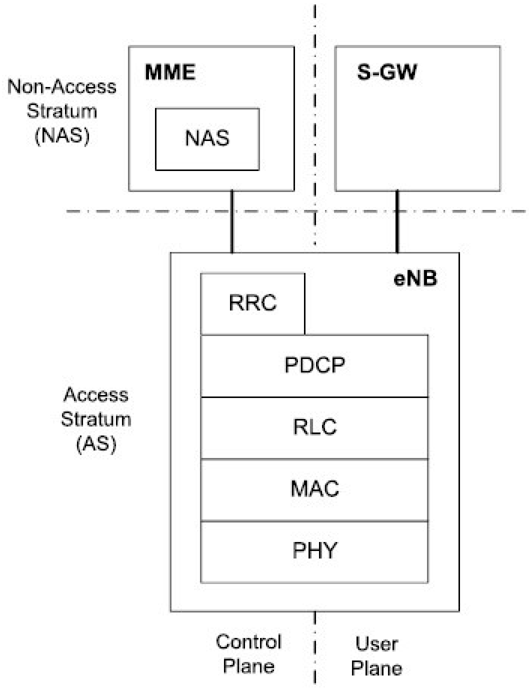


Figure 2.5: LTE’s Protocol Stack. Extracted from [8].

- **PHY:** the Physical Layer is responsible for handling the Orthogonal Frequency-Division Multiple Access (OFDMA) operations, the modulations and the coding of different control and traffic channels. It incorporates reference signals which are used for channel estimation and equalization;
- **MAC:** the Medium Access Control is in charge of the multiplexing/dedmultiplexing of RLC Packet Data Units (PDUs), the scheduling information reporting, error correction through Hybrid ARQ, local channel prioritization and padding;
- **RLC:** the Radio Link Control makes error correction through Automatic Repeat reQuest (ARQ), segmentation according to the size of the transport block and protocol error detection and recovery;
- **PDCP:** the Packet Data Convergence Protocol is in charge of header compression, in-sequence delivery and re-transmission of PDCP Session Data Units (SDUs) for acknowledge mode radio bearers at handover and ciphering and integrity protection;
- **RRC:** the Radio Resource Control is in charge of broadcast system information related to Non-Access Stratum (NAS) and Access Stratum (AS), establishment, maintenance, and release of RRC connection, security functions including key management, mobility functions, QoS management functions, UE measurement reporting and control of the reporting and NAS direct message transfer between UE and NAS;
- **NAS:** the Non Access Stratum is responsible for connection/session management between UE and the core network, authentication, registration and location registration management.

2.1.4 LTE Radio Interface

In this section the main mechanisms of the LTE radio channel will be explored. We will briefly discuss OFDMA, MIMO and resource blocks.

OFDM was one of the technologies that helped increase the spectral efficiency of modern LTE. This technique works by dividing the available bandwidth in a group of orthogonal sub-carriers. In Figure 2.6 it is represented a single sub-carrier on the left and on the right the full OFDM signal. As one can see, these sub-carriers are generated in a way that in every peak of any of them in the frequency spectrum, the contribution of all the others sub-carriers is zero. This technique allows a dramatically reduction of the guard band between the sub-carrier's frequencies.

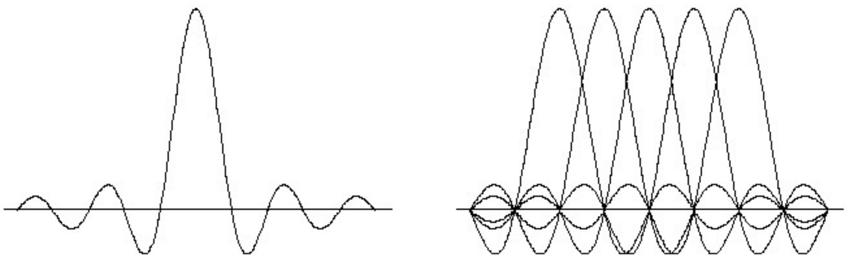


Figure 2.6: Single Carrier (Left). Full OFDM signal (Right). Extracted from [9].

This modulation transmits smaller symbols in each carrier and with a longer duration in time. Overall, this modulation achieves a higher bandwidth efficiency because it sends several symbols at the same time, and each transmitted symbol has a higher endurance against bad environment conditions, such as high frequency attenuation, inter-symbol interference and multipath interference.

LTE uses OFDMA in the DL, a variation of OFDM, where one does not allocate all the sub-carriers to the same user, instead, the sub-carriers are distributed between the users. This means, that in LTE, users are assigned resources in the time and frequency domains.

In the UL LTE uses Single Carrier Frequency Division Multiple Access (SC-FDMA). We can see this as a special case of OFDMA where the signal is pre modulated before being modulated in OFDMA. Each UE pre modulates their signal with a silent Fourier coefficient and the final result is a single carrier signal. The reason this type of modulation is used in the uplink lies in the fact that the resulting signal has a lower Peak to Average Power Ratio (PAPR), allowing for the UE to save energy resources.

Another technique that is broadly used in LTE is MIMO. This means that instead of having only one antenna, there will be multiple antennas that can be used for different purposes. As illustrated in Figure 2.7, MIMO can be used in 4 configurations to improve different channel characteristics.

Firstly, it can be used to increase signal diversity, where the multiple antennas transmit the same signal to the same destination. This configuration allows a more reliable and error free communication with the destination. Secondly, it can be used for multiplexing, where the two antennas send different signals to the same destination in order to increase the final data rate to the destination. Thirdly, it can be used in Space Division Multiple Access (SDMA), where the two antennas send different signals to different users. This configuration increases the capacity of a cell. Lastly, it can be used for beaming, where the antennas emit special signals that, when combined, allow to increase the coverage of the cell.

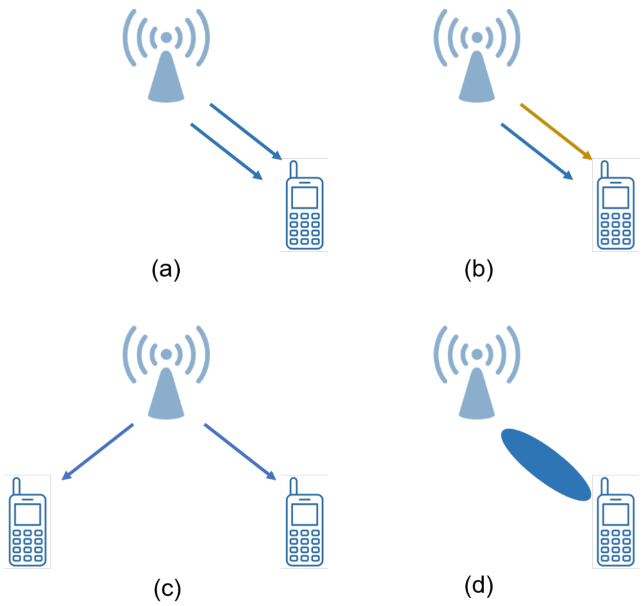


Figure 2.7: MIMO Configurations. (a) Diversity for more reliable communications, (b) Multiplexing for higher data rates, (c) SDMA for improved cell capacity, (d) Beaming for improved coverage. Adapted from [6].

MIMO initially was used in a 2x2 antennas configuration, but currently there are commercial applications with a greater order of MIMO. With the arrival of 5G it is expected that Massive MIMO becomes available. This advance is already being deployed, and enabling systems to use up to 96 or 128 antennas.

Resource allocation in LTE is based on Resource Blocks (RB), which is the minimum allocation a UE can get. Figure 2.8 shows the structure of a RB. The smallest unit of a RB is the Resource Element (RE) which translates to using one sub-carrier (15 kHz) to transmit one radio symbol. One RB uses 12 sub-carriers during 0,5 ms. In 0,5 ms it is possible to transmit 6 or 7 symbols, depending on the cyclic prefix, which leads to one RB having $7 \times 12 = 84$ symbols.

Depending on the modulation used, one symbol is equivalent to transmit two, four or six physical channel bits. The modulation is chosen taking into account several factors, the most important being the quality of the physical channel. The Base Station (BS) chooses the modulation based on the Channel Quality indicator (CQI) sent by the UE, which is selected taking into consideration the radio channel, but also the UE's capacity to correct the signal it gets. This means that a UE in the same radio conditions as another may have a more efficient correcting algorithm, allowing it to announce a higher CQI. In the Table 2.1 it is the modulation used in each CQI.

Since every mobile operator has to adapt the standards to their specific environment, LTE defined several supported bandwidths described in the Table 2.2. One may notice that in every configuration operators have to leave a guard band on the limits of the bandwidth. This guard band is used to prevent interference between spectral domains.

LTE can be deployed using various Radio Frequency (RF) channel configurations. There are 27 bands specified for Frequency Division Duplex (FDD) and 12 bands for Time Division Duplex (TDD). LTE can coexist with the previous 3GPP technologies, so some bands can be used simultaneously by

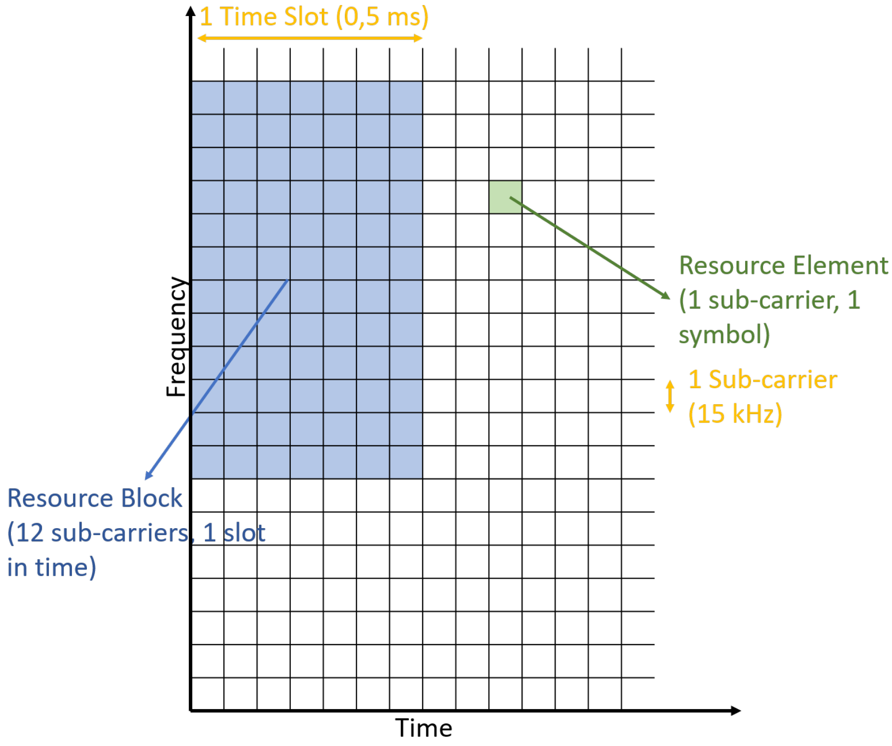


Figure 2.8: Structure of a Resource Block. Adapted from [10].

Table 2.1: CQI Index.

CQI Index	Modulation	code rate $\times$ 1024	Efficiency (Bits per Symbol)
0	-	-	-
1	QPSK	78	0.1523
2	QPSK	120	0.2344
3	QPSK	193	0.3770
4	QPSK	308	0.6016
5	QPSK	449	0.8770
6	QPSK	602	1.1758
7	16QAM	378	1.4766
8	16QAM	490	1.9141
9	16QAM	616	2.4063
10	64QAM	466	2.7305
11	64QAM	567	3.3223
12	64QAM	666	3.9023
13	64QAM	772	4.5234
14	64QAM	873	5.1152
15	64QAM	948	5.5547

LTE and by other technologies. In Europe, mobile operators have over 600 MHz of spectrum available for LTE, including the 800 (LTE800), 900, 1 800 (LTE1800), 2 100 and 2 600 MHz bands (LTE2600).

Table 2.2: Cell Bandwidth.

Total Bandwidth	#Resource Blocks	#Sub-carriers	Occupied Bandwidth	Usual Guard Bands
1.4 MHz	6	72	1.08 MHz	2×0.16 MHz
3 MHz	15	180	2.7 MHz	2×0.15 MHz
5 MHz	25	300	4.5 MHz	2×0.25 MHz
10 MHz	50	600	9 MHz	2×0.5 MHz
15 MHz	75	900	13.5 MHz	2×0.75 MHz
20 MHz	100	1200	18 MHz	2×1 MHz

In Portugal, ANACOM issued an auction for some of the listed bands to be used to deploy LTE. The three major network operators in Portugal (MEO and NOS, Vodafone) bought the rights to use 29 of the 39 lots under auction. Table 2.3 compiles the bands each operator owns. These values take the 6 MHz bandwidth that some operators already had in the 1 800 MHz band for GSM1800 into account. In Europe, FDD is the most used duplex mode, so the frequencies presented in Table 2.1 are related to this mode.

Table 2.3: Frequency Bands Owned By Operators. Compiled from [11] and [12].

Band [MHz]	**Owner**	**Total Spectrum [MHz]**	**Uplink Band [MHz]**	**Downlink Band [MHz]**
800	MEO	2 x 10	832 – 842	791 – 801
	Vodafone		842 – 852	801 – 811
	NOS		852 – 862	811 – 821
1800	MEO	2 x 20	1 710 – 1 730	1 805 – 1 825
	Vodafone		1 730 – 1 750	1 825 – 1 845
	NOS		1 750 – 1 770	1 845 – 1 865
2600	MEO		2 510 – 2 530	2 630 – 2 650
	Vodafone		2 530 – 2 550	2 650 – 2 670
	NOS		2 550 – 2 570	2 670 – 2 690

Lower frequency bands are currently used by mobile operators to provide LTE coverage. On the other hand, higher frequency bands have a small coverage area due to the fact that signal attenuation increases with frequency. However, these bands are also important, because they are used to increase network capacity. In the specific case of LTE in Portugal, the 2 600 MHz and 1 800 MHz bands provide up to 20 MHz or 14 MHz of additional capacity, respectively.

2.1.5 LTE Scheduler

This section also considers the information in [13].

One important component of the LTE system is the scheduler. Since in LTE resources can be allocated both in frequency and time domains, this component is responsible for distributing the RBs by the UEs. In the Figure 2.9 it is represented an example of how the resources can be allocated.

This component is not standardized, which means that each operator defines the way they want to distribute the resources. Despite the operators being able to choose how RBs are allocated, these allocation policies can be divided in three types:

- ***Type 0:*** Policies are of type 0 when the BS collects the resource blocks into Resource Block Groups (RBG), which are assigned individually using a bitmap;
- ***Type 1:*** These policies can assign individual resource blocks within a group, but have less flexibility over the assignment of the groups themselves. Allocation type 1 might be suitable in environments with severe frequency-dependent fading, in which the frequency resolution of type 0 might be too coarse;
- ***Type 2:*** This type is used when UEs require a constant number of RBs for a certain amount of time, the BS gives a continuous allocation of Virtual Resource Blocks (VRBs).

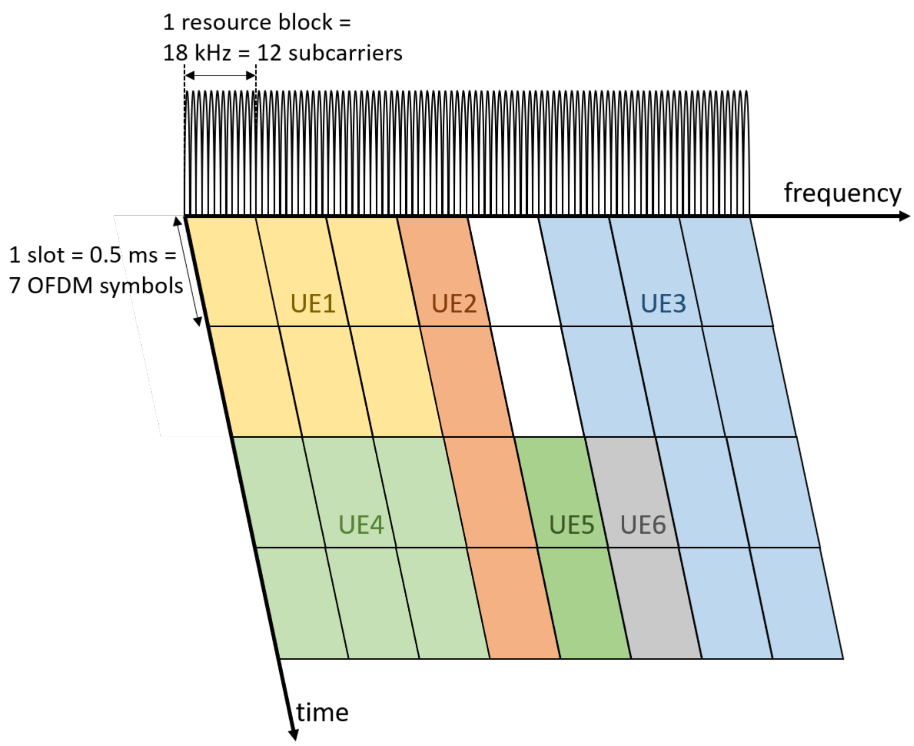


Figure 2.9: Resource Block Allocation. Adapted from [10].

These are some examples of generic schedulers:

- ***Maximum Throughput Scheduler:*** The network tries to exploit channel variations to maximize the cell throughput. This scheduler starts by satisfying completely the users who have the best channel conditions. As a result of this policy, the users end up with very unbalanced resource allocation, which may lead to a starvation of some users.
- ***Blind Equal Throughput Scheduler:*** This policy distributes the resources in a Round Robin approach. The main disadvantage of this scheduler is that it distributes the resources in a very

inefficient way. This scheduler also creates large discrepancies in the service distribution. Users with the same number of resource blocks but different distances to the BS will get very different throughputs.

- ***Proportional Fair Scheduler:*** This scheduler tries to balance the two previous schedulers by allocating resources taking into account both the maximum rate achievable by a UE, and the average rate the UEs get. This scheduler tries to achieve a situation where all users end up with a throughput closer to the global average.

Mobile operators can then build more complex and proprietary allocation policies trying to improve user throughput, user delay, user fairness and other operator specific metrics. The scheduler takes as input several indicators about their UEs in order to make the decisions about resource allocation. One of those indicators is the CQI which, as discussed, is also used to define the signal modulation.

2.1.6 Capacity

Considering resource allocation is organized in RBs, this is also the way LTE's capacity is measured. Moreover, to translate the number of RBs available into the number of users the system can serve it is important to know how many resource blocks each user needs. The number of RBs available is dependent on the available bandwidth (Table 2.2), which is a result of assuming each RB is composed of 12 sub-carriers and each sub-carrier occupies 15 KHz. An approximate relation between LTE's bandwidth and the number of resource blocks is given by the expression:

$$N_{RB} = \frac{B_{ch[kHz]}}{B_{RB[kHz]}} \times \frac{P_{Bch[\%]}}{100}, \tag{2.1}$$

where:

- N_{RB}: number of RB;
- $B_{ch[kHz]}$: channel's bandwidth ;
- $B_{RB[kHz]}$: RB's bandwidth, which is 180 kHz;
- $P_{Bch[\%]}$: channel's bandwidth used percentage, $\sim 90\%$.

Once the number of RBs each user gets is known, we have to assess the amount of information each resource block can transport to compute the throughput. This quantity depends on the modulation used, which depends on the quality of the radio channel. This constrains will result in a different spectral efficiency for every user. Claude Shannon defined a theoretical maximum for the spectral efficiency:

$$R_{b,max[bit/s/Hz]} = log_2(1 + \rho_N), \tag{2.2}$$

where:

- $R_{b,max[bit/s/Hz]}$: maximum data rate per hertz;

- ρ_N: SNR (in linear units).

This is a theoretical value. Figure 2.10 shows the exact relation between the SNR and the obtained throughput depending on the used modulation. As expected, each modulation requires a minimum level of SNR to be usable, and none of the modulations can reach the Shannon curve (in black), which is the theoretical value we use.

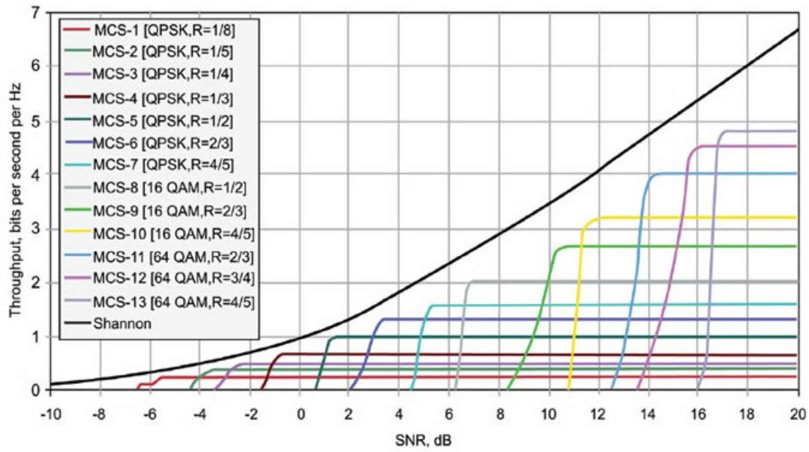


Figure 2.10: Throughput per SNR by modulation. Retrieved from [14]

.

2.1.7 Services and Performance Parameters

Mobile networks give support to many services and the throughput requirements are very different from one to another. In the Figure 2.11 there is an illustration of the prevalence of these different services. The high demand for services that consume a big amount of data was already clear in 2017 with video being 55% of all mobile data consumption, however, Ericsson predicts that this trend will keep increasing up to 75% in 2023.

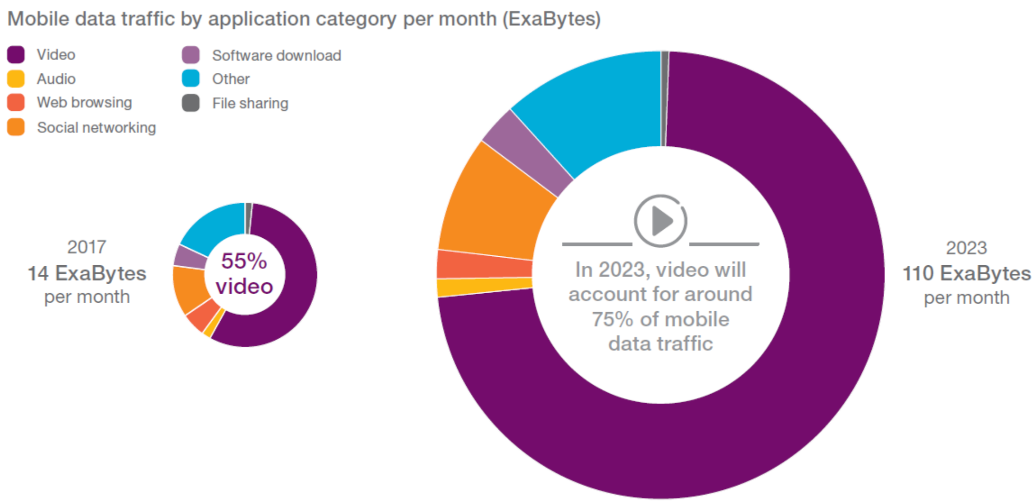


Figure 2.11: Traffic by service. Retrieved from [1].

3GPP divided services into 4 classes: Conversational, Streaming, Interactive and Background. Table

2.4 presents the main characteristics/requirements of the services in these categories. These classes are used to prioritize the data flows, allowing eNBs to treat data traffic differently, even if it is originated in the same device.

Table 2.4: Service Requirements. Adapted from [6].

Service	Real-Time	Symmetric	Delay	Buffer	Example
Conversational	Yes	Yes	Minimum, Fixed	No	Voice
Streaming	Yes	No	Minimum, Variable	Yes	Video
Interactive	No	No	Moderate, Variable	Yes	Web Browsing
Background	No	No	High, Variable	Yes	E-mail

A good network planning should take these categorization into account, studying what kind of services will be more required by the users. One way to predict the different demands is to assess what kind of mobile terminals will be more used in a cell. For example, if the cell is near a stadium or a garden, probably there will be more mobile phones than tablets, if it is an office area probably there will be a large amount of computers using mobile services. Since some kinds of services are much more used in certain devices than others, it is possible to make a global estimation of service usage in a cell. Figure 2.12 is a good illustration of the different consumer habits by device.

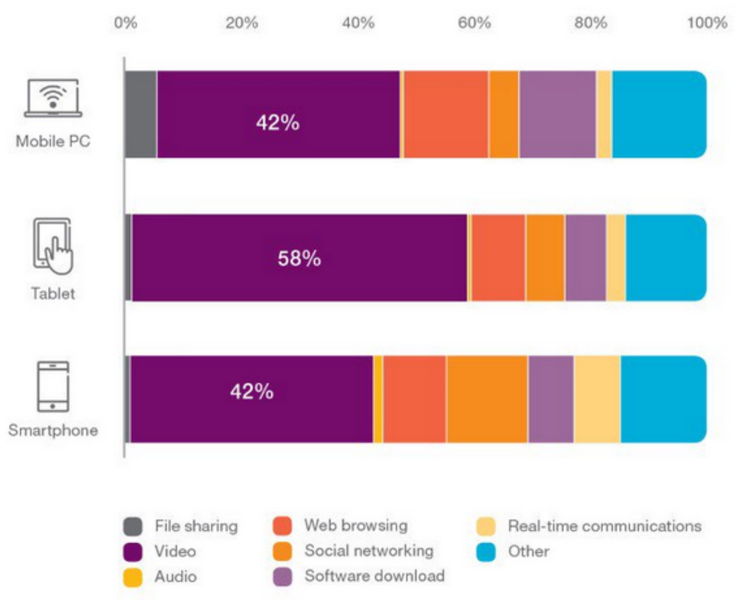


Figure 2.12: Average mobile data traffic volumes by application category and device type. Retrieved from [15].

Once the distribution of demands by service is estimated, it is important to know exactly how much these services will be consuming. In the Table 2.5 there are some examples of some services, their service class, and their approximate minimum, medium and maximum throughput.

Table 2.5: Throughput By Service. Extracted from [16].

Service		Service Class	Throughput [kbps/s]		
			min	**med**	**max**
VoIP		Conversational	8	32	64
E-mail		Background	10	100	1333
M2M	Smart Meters	Background	-	200	-
	e-Health	Interactive	-	200	-
File Sharing		Interactive	200	1600	4 444
Video Streaming		Streaming	500	5000	10 000

2.1.8 Relaying

Relaying is the technique in which by the addition of network nodes we complement the network of eNBs. This allows expansion of coverage or increase of capacity. Early relays were functioning as simple repeaters. Compared with the deployment of a full base station they had lower costs, since they involved no baseband processing, no backhaul network installation nor subscription fees for access to the fixed public network. However, they presented some disadvantages: they repeated and amplified interference, and, since they operated independently from the radio access network, they required a separate Operation and Maintenance functionality.

Modern relays in 4G designated Relay Nodes (RNs), are network nodes connected wirelessly to a source eNB. Since they are under the full control of the radio access network, they allow similar monitoring and remote control capabilities as for an eNB. The drawback of this control is the additional delay introduced.

Figure 2.13 shows the terminology relating RNs that 3GPP introduced:

- **Donor eNodeB/cell:** The source eNodeB/cell from which the NR receives its signal.
- **Relay cell:** The coverage area of the RN.
- **Backhaul link:** The link between the donor eNodeB and the RN.
- **Access link.** The link between the RN and a UE.
- **Direct link.** The link between the donor eNodeB and a UE.
- **Inband/outband.** An inband RN uses the same carrier frequency for the backhaul link as for the access link; otherwise, the RN is said to be outband.
- **Half/full duplex.** A half-duplex RN cannot receive on the backhaul link at the same time as transmitting on the access link, and vice versa, whereas a full-duplex RN has sufficient antenna isolation to be able to operate without this restriction. This distinction applies to inband RNs only, since outband RNs are always full-duplex.

- **Donor and coverage antennas.** At the RN, the donor antenna(s) are used for the backhaul link, while the coverage antenna(s) are used for the access link. In some cases, the physical donor and coverage antennas may be the same.

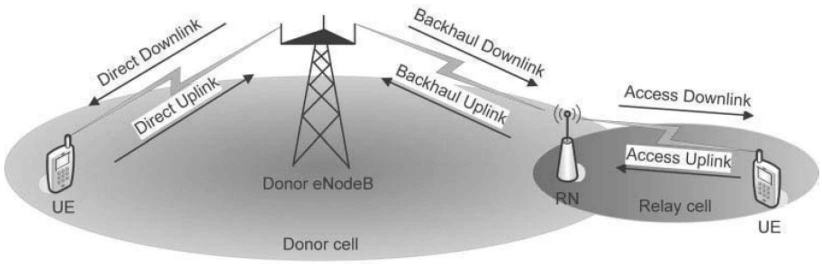


Figure 2.13: Relay Terminology. Extracted from [2].

Backhaul's link on inband RNs consume radio resources, thus reducing the capacity of the RN. When RNs operate in half-duplex, there are also other complications for the system design. On the other hand, outband relaying with full-duplex operation increases the relay cell capacity and simplifies the system design. This comes at the expenses of the system cost since the last configuration requires two isolated antennas, and the availability of a second carrier frequency.

Overall, the main benefits of installing a RN over an eNB are the reduction of the infrastructure and the operational costs. RNs do not require any wired or microwave backhaul connection to the network, avoiding a significant part of the network's CAPital EXPenditure (CAPEX) and OPerational EXPenditure (OPEX).

RNs can belong to one of three categories, depending on the functionality layers they provide. Layer 1 RNs are simple repeaters and may include some baseband processing such as Forward Error Correction (FEC). Layer 2 RNs provide Medium Access Control (MAC) functions such as scheduling. Layer 3 RNs have their own PCI signalled by the PSS/SSS, and all of Layer 1 and Layer 2 functions are supported by the RN.

In Figure 2.14 are illustrated examples of some use cases for relaying. Relays can be used for cell coverage extension, where the RN is placed near a cell coverage limit to extend the coverage of the cell; they can be used for outdoor capacity boost, where the RN is placed inside the cell limits to provide a capacity boost in particular areas of the cell (hotspots), in which there are an unusual number of users; they can be used to provide indoor coverage enhancement where the RN is placed inside buildings; they can be used for dead spot mitigation, where the RN is used for filling coverage holes inside the macro network; they can be used for temporary deployments to provide an extra coverage or capacity in situations like special events or emergency deployments; and finally, they can be used for group mobility, where the RN is placed inside a bus or a boat to provide service to passengers. Since the RN is capable of using a more powerful signal, operations like handover are more efficient and users spend less battery on their devices.

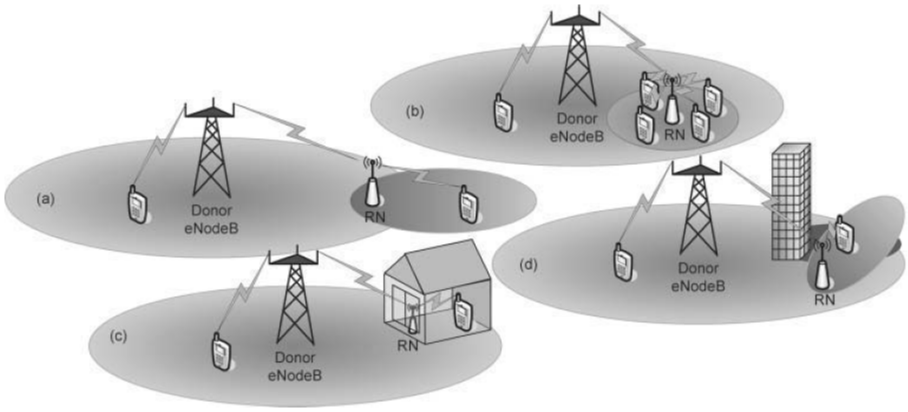


Figure 2.14: Relay use cases: (a) cell coverage extension; (b) capacity boost; (c) indoor coverage enhancement; (d) dead spot mitigation. Extracted from [2].

2.2 UAVs

This section will address UAVs. Beginning with some general considerations about the topic, then proposing a categorization for these devices and finally explaining how they can be applied in a telecommunications scenario.

2.2.1 UAVs General Considerations

UAVs have been seeing a rapid technological advancement in the last years. This is largely explained by the high amount of applications these devices can have. Usually UAVs are adapted to the specific task at hands, which makes them highly specialized and efficient tools. This is the reason why there are so many different types of drones. Recently UAVs have become object of high interest for the general population due to their recreational utilities, and for businesses because of their versatility. This has caused a boom in the UAV market, banalizing the technology.

UAVs are being studied and built since as early as prior to the ending of the World War I [17]. Although at this point a remote pilot was still necessary. Nowadays, almost all the control functions of an UAV can be delegated to an auto-pilot.

Modern UAV controller systems are turning possible the collaboration between several drones, making them aware of neighboring drones so they can work together to serve a joint purpose. Depending on the complexity of the collaboration, some centralized controller may be needed.

The deployment of UAVs in different telecommunications scenarios seems promising. Harvard Business Review [18] launched a video giving some examples of areas where they believe UAV deployment is promising to reduce operational costs and times. The examples portrait uses going from cell tower inspections, to deployment of cell-drones to increase mobile network operation.

2.2.2 UAV Types

In [19] authors revised the literature on UAV, creating a comprehensive revision of literature on this topic. Although categorizing drones is not easy because of their very disparate characteristics, they were able to categorize UAV into two important classes: Heavier-than-air, and Lighter-than-air. The first class was then divided by Wing types and Rotor types. Figure 2.15 shows this categorization. The next paragraphs notes the main differences between these different types.

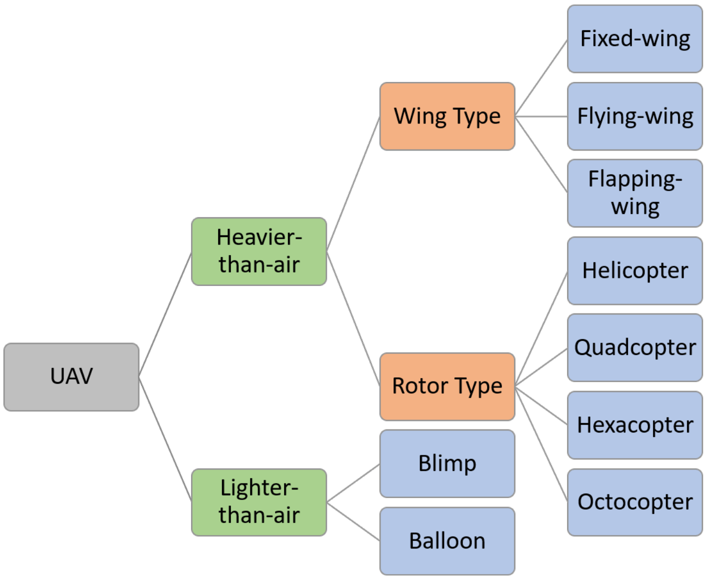


Figure 2.15: UAV Types. Adapted from [19].

All the rotor type UAVs use rotors to propel themselves in the air. They differ in the amount of rotors, which results in different degrees of stability, rotor redundancy and in the amount of time they can stay in the air due to battery capacity. Comparing to the other categories they are one of the most agile type of drone, capable of changing direction and speed very fast, but also the ones that consume more energy to stay in the air.

The fixed-wing UAVs, compared to multi-rotor drones, are a lot more energy efficient and can glide even after their rotors stop. This makes them safer and capable of maintaining themselves in the air for a longer period of time. However, this type of UAV requires a larger space for take-off and landing, and they are unable to hover over a specific location.

The flapping-wing UAVs are usually a hand size. They generate lifting and forward force by flapping their wings in an bird like way. However, usually they are not able to carry a lot of weight.

The Lighter-than-air type of drone uses a compartment filled with hot hair or helium to generate lifting force. There are several systems that can be applied in this kind of drones to give them manoeuvrability. Among them are rotors and ionic propellers.

2.2.3 Applying UAVs in Telecommunication Networks

Applying UAVs to telecommunications seems to be a logical step, in [20] the authors studied the use of UAVs in emergency scenarios. Although they conclude that UAVs are generally a good fit for these scenarios, they raise relevant questions about power consumption in these peculiar situations where basic infrastructures may be compromised.

In [21] the authors studied the relaying in tactical situations, proclaiming increases by a factor of two in throughput, and by 67% in connectivity when compared to the ad hoc ground network without using UAVs.

In [22] and [23] the authors focused on studying the impact of altitude in the UAV performance assuming the drone-cells are isolated from other base stations. The altitude of the drone is responsible for an important trade-off between achieving a Line of Sight (LOS) radio channel with the users, which is more probable with higher altitudes, and achieving a lower path loss, which is lower at smaller distances, hence, lower UAV's altitudes.

In [24] the integration of drone-cells in the existing LTE network is addressed, making the case that this is not a simple task, and ending up suggesting a multi-tier approach to this problem where high level UAVs, flying at a higher altitude, give support to lower level UAVs. They also envision a decoupling of this activity (UAV deployment) from the current network providers, and becoming themselves infrastructure providers and mobile virtual network operators providing services to the current network operators.

In [25] a clustering algorithm of type K-Means is used for addressing the problem of UAV localization in a scenario where UAVs and fixed based stations coexist. This paper simulates, discusses and compares scenarios where there are only macrocells, planned or random picocells, and finally a scenario where UAVs are deployed alongside macrocells.

2.3 Genetic Algorithms

A GA is a meta-heuristic algorithm. These type of algorithms are generally used to solve optimization problems. They use a combination of random choices and knowledge of previous results to address the search space. GA, more specifically, is a type of algorithm that takes advantage of the principles of nature's genetic evolution to do this. This type of algorithms have been being studied since the 1950s. Since they were discovered they have been used to solve a wide range of optimization problems. This section is based in [26] and [27].

2.3.1 Components

Many of the mechanisms used in these type of algorithms are stochastic, which presents a challenge to the analysis of those mechanisms. There are many decisions relating the implementation of such algorithms that divide the scientific community.

The higher level component of genetic evolution is the generation. Generations are composed of individuals. Those individuals have genetic material that determines the characteristics/traits of each individual. This genetic material is organized into chromosomes and genes, one individual has one or a group of chromosomes, which are composed of genes. The gene is the most basic component of genetics, this structure stores the information about one specific trait of its individual. Making an analogy, if it were a human, the individual would be a specific person, and a gene would encode a specific trait of that person, for example it's hair color. According to the value in that gene the person would be blond, brunette or red haired.

In order to develop a specific genetic algorithm, we have to be able to frame our problem in a way that we can represent it with these basic components. We must decide which part of our problem will be the individual, what will be a chromosome, how many will there be and what information will encode each gene. This task should not be underrated since it is known that the chosen representation will directly impact the performance of the algorithm. Given the same problem, different authors will advocate for different representations for the sake of two things: convergence speed and the ability to avoid locally optimal solutions. This trade-off will be better explored further ahead, but sometimes it is as hard to choose the best representation as it is to solve the original problem.

2.3.2 Evolution

Before one can understand how the evolution to the optimal solution works, one needs to understand what is fitness. In nature, fitness is defined as the level of adaptation of a certain individual to its environment. In the end, the more fit an individual is, the more likely it is to have a longer life, to reproduce and to generate descendents. It is called natural selection to the process of elimination of lesser fit individuals, although to be fair, they are not exactly eliminated, they simply are not chosen to reproduce, and so in the coming generations, their specific combination of characteristics seizes to exist. In the genetic algorithms world, the fitness value is a representation of how well a certain solution responds to the initial problem. The bigger its fitness value the closer it is to being the optimal solution. The design of this ranking tool, is one of the most difficult challenges in designing GAs.

The main genetic operators are the following:

- **selection**, how we choose individuals to reproduce;
- **crossover**, how does the mixing of genetic material between two parents happen;
- **mutation**, the random process of changing gene values.

All of these processes represent trade-offs that will be explored further ahead. That said, it is important to notice that they are non consensual, and each one of these processes can be, and is still, object of study by the scientific community.

Figure 2.16 shows the genetic evolution process which starts with a random group of individuals, the first generation. Then we iterate the following process to generate the subsequent generations. Choose two individuals from the previous generation, crossover their genetic material to generate a new individual, randomly mutate its genes and add this individual to the new generation. Every generation has the same number of individuals so we iterate this process until the next generation has the same number of individuals as the first generation.

To fully understand the subtleties of the referred operators we should better understand what is convergence speed and local optimum in the context of genetic algorithms. The convergence speed is

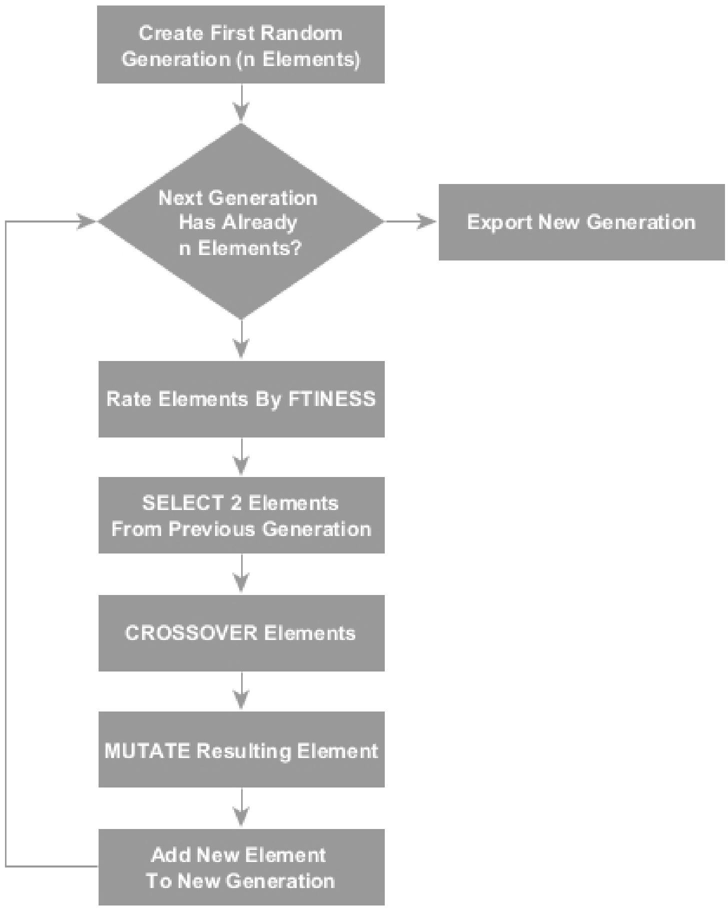


Figure 2.16: Evolution Process.

the required number of iterations/generations we need to reach a stable value of fitness. In theory, one can use this parameter to know when to stop iterating generations. In every generation one can search for the best adapted solution, and if its value of fitness is unchanged for a large number of iterations one can assume the optimal solution has been reached and stop iterating. Putting aside for a moment other important factors, the time needed for this to happen is directly connected to the search space being addressed. The larger the number of possible options for an individual, the longer it will take to reach this optimal solution. Returning to the human analogy, this is the same as saying that the bigger the color palette for hair color, the more iterations will be needed to reach the final solution.

Besides the convergence speed to the solution, it is important to reflect on the quality of the solution one gets. In general, there is no certainty that a genetic algorithm will find the best solution. By definition this is a stochastic algorithm, and so, it is possible that the algorithm converges to a local optimum by chance and is not able to detect that it is in a local optimum. However, there are ways to improve the odds of finding this perfect solution. One of them is to make an effort to design the genetic operators in a way that the generations are kept with a fair amount of diversity, making sure that solutions with a low value of fitness are not deleted as soon as they appear because they may be, globally, a bad fit for the problem, but they may contain some of the genes present in the global optimum.

2.3.3 Operators

The next subsections will address the various operators present in the a genetic algorithm.

Selection Operator

The selection operator intends to simulate the mating behavior of the individuals. This process is designed to distribute the selection probabilities to the different individuals. In nature, the more fit an animal is, the more likely it is to be chosen by a different gender individual to mate. The problem is that there is no absolute rule to chose a matting partner in nature. Moreover, that seems to be a good thing overall, since this allows for lesser fit animals to reproduce anyway, keeping the generations with a relative degree of variability. This variability seems to be one of the main factors behind adaptation to an ever changing environment.

To create a new generation, we need to choose pairs of individuals of a previous generation to crossover and generate a new individual. There are several ways we can use to select the elements to be used in this process. According to [28] it is possible to name some categories for this selection methods. These categories define how one should distribute the selection probabilities.

Taking that into consideration, a selection method can be *dynamic*, meaning the selection probabilities are proportional to the actual fitness-values; or it can be *static* meaning the individual is selected according to its position in the fitness ranking.

A selection method can be *preservative*, which attributes a non-zero probability to every element in a generation, meaning that every element has the possibility to reproduce; or *extinctive* which dictates that some elements are excluded right away from the reproduction operation.

When a selection method is *extinctive*, it can be *left* or *right* extinctive. If the elements with zero probability are the low performers, we are in the presence of a right extinctive selection. If the elements with zero probability are the top performers we are in the presence of a left extinctive selection. Although this last category may seem counter productive, it is very useful if one wants to make sure that the generations are kept with a certain degree of variability, thus preventing the algorithm to converge too soon to a local optimum.

Aside from the distribution of probabilities, it is possible to define a selection method as being *elitist* or *pure*. In the elitist method, some of the top performers skip the reproduction process and are directly included in the next generation.

Fitness Function

In nature, the fitness of an individual is its capability of enduring its environment. In genetic algorithms, the fitness function is the ranking tool used to distinguish the individuals that better answer to the initial problem, from those who do it worse. The more accurate the fitness function is, the best our final solution will be.

There are several phenomenons that can turn a fitness function inaccurate. In some situations there may be no known fitness function that correctly assesses the individual's fitness; some simplifications of the fitness function may be required in order to keep its complexity between some defined limits; noisy information can disturb the accuracy of the fitness function.

Crossover Operator

Another important aspect of the reproduction is how the genes of the two parents are mixed up in order to generate offspring. It is important not to forget that the issues surrounding this operation are different whether a binary representation is used or not.

When a binary representation is used, the simplest way to implement this operation is to choose a random point in the chromosome and all of the genes before that point are copied from one parent, and all the others are copied from the other parent. This implementation is overall correct, however, there are some things to consider, since there are some combinations that are not possible with this implementation. For example, it is not possible to combine 11*****1 with ****11** to create a 11**11*1. Another effect of this implementation is that some genes are treated differently from all the others. For example, every segment exchanged has the top genes of the parents. This means that *a priori* one can tell that the first gene will be from one parent, and the last is from the other. This is the reason some people choose to implement a two point crossover operation. In this case, two points of the chain are chosen and the segments between them are exchanged. Lastly, it is important to check if it makes sense, in a specific implementation, to allow a gene to be cut in half in this operation.

Although these considerations are specific to a binary representation, some of them are still valid in other representations. In these other representations there may be specific considerations to make, there may be the case where a specific crossover creates a descendent that doesn't make sense, or it is not valid. In these cases, the crossover operation may be a very specific and different operation.

Mutation Operator

The last important operator in GA is the mutation. Along with crossover it is the main instrument of disruption and innovation in an evolution process. This operator is one of the things that keeps the evolution process from finding a local optimum. The balance of amount of mutation and crossover is extremely important to the convergence speed/global optimum trade off.

2.3.4 Stop Conditions

There is one more factor of GAs that needs to be addressed, which is the stopping condition. There are several ways that can be used to define this condition. Firstly, one can impose a limit to the number of iterations the algorithm runs. Then, one can impose a limit to the time the algorithm is running. And finally the algorithm can stop when the chance of achieving a significant change in the fitness values is very low.

The first two options require a knowledge about the problem to allow the estimation of a reasonable maximum number of iterations or time. This means that these values will vary for each GA.

The third option is adaptive, this means that depending on the specific GA it will yield a different stop condition. However, determining the chance of achieving a significant change is also not an easy task.

According to [29] the concept of convergence can be divided in two. In can be genotypical or phenotypical. In the first, the alleles are evaluated, and if ϵ % of the population has the same value on that allele, they consider the population to have converged and stop iterating. Of course ϵ has to be high and pre-determined. In the second the stopping condition uses, for example, the fitness value, which is dependent on the phenotype, and when it reaches a pre-determined ϵ it stops iterating.

2.4 K-Means Drone Disposition Algorithm

K-Means is a well known clustering algorithm, which works by minimizing the distance of all the points/UEs to the clusters' centroids coordinates. However, it is possible hat this algorithm finds local optima, and there are several factors that can contribute to making this local optima as close as possible to the global optimum. That is the reason why there are some possible variations in its implementation. One of the things that is known to affect the probability of finding the optimal solution is the initialization. For this reason this work implements a variation of K-Means called K-Means++. This algorithm will be used as a baseline for the GA, and the specific implementation details were based in [30], [25] and [31].

This solution wishes to assign all UEs in U to K clusters, where U are the coordinates of the UEs $U = \{u_1, u_2, ..., u_n\}$. Given a set of points $X = \{x_1, x_2, ..., x_n\}$, a Voronoi region is defined as

$$r_i = \{w \in W :\parallel w, x_i \parallel < \parallel w, x_j \parallel \forall j \neq i\}, \tag{2.3}$$

where $\parallel . \parallel$ denotes the Euclidean distance, and W is the set of points that make up our scenario. The UE cluster $s_i \subset U$ is the set of UEs which are located inside of the associated Voronoi cell

$$s_i = U \cap r_i. \tag{2.4}$$

The centroid of s_i is defined as the mean coordinates of the UEs belonging to the cluster, which is defined as

$$\mu_i = \frac{\sum_{u \in s_i} u}{\mid s_i \mid}. \tag{2.5}$$

The goal is to select the points X which generate Voroni regions and UE clusters such that

$$min \sum_{i=1}^{K} \sum_{u \in s_i} \parallel u, \mu_i \parallel, \tag{2.6}$$

The general flow of the algorithm is depicted in Figure 2.17. The major difference between K-Means and K-Means++ is in the initialization. While in the first the initial position of clusters is chosen at random, in the second they are chosen in a way that they are the farthest away possible from each other. To achieve this result, the first cluster takes the coordinates of one random point. Then, we recursively choose the point that is farthest away from any cluster to be the next cluster.

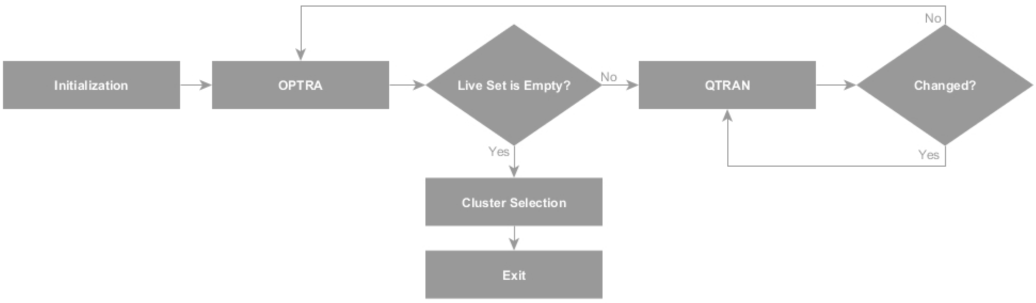


Figure 2.17: K-Means++ high level architecture.

One of the parameters defined in the initialization process is the number of clusters. Since the end goal is to use the cluster's coordinates for drone positions, we will try to create clusters with a number of UEs which drones can fully serve. Since K-Means generates clusters with expected N/K UEs, where N is the total number of UEs and K is the number of clusters, then $K = N/M$, where M is the drone's capacity. Let D be the number of available drones, since $D \neq K$ a final step will be required to select the clusters that will be used as drone coordinates.

After initialization, K-Means iterates through all UEs recursively and allocates UEs to the closest cluster, every time an allocation like this takes place, the coordinates of the clusters are updated to be the mean value of the coordinates of all it's UEs. This process is represented in Figure 2.17, but is divided into two sub-processes: the Optimal-Transfer (OPTRA) and the Quick-Transfer (QTRAN). The detailed steps of this two processes can be found in [30], but in short, in the OPTRA stage the position of UEs is compared against all clusters and in the QTRAN the UE's position is only compared with the two closest clusters.

In the Figure 2.18 we can see the evolution of the algorithm. Each color represents a cluster, the smaller dots are the UEs and they are represented in the color of their final cluster. Each big dot represents the position of the cluster's centroid in one of the iterations. The last iteration, with the final positions, has a black square on it.

Returning to Figure 2.17, the last step of K-Means++ algorithm before terminating is the selection of the clusters to become drone's positions. This process is necessary because it is assumed the algorithm will be used in a situation where it is not possible to offload all UEs from the macrocell to the drones. If that is not the case, simpler algorithms could be used. In order to maximize the efficiency of the drones, we should select those clusters serving more UEs, and those furthest away from the macrocell. This is the same as saying that since we are spending resources in deploying a drone, we want it to be used at full capacity, either serving more UEs or providing a better signal to UEs further away from a macrocell.

To achieve this goal, the clusters must be ordered by relevance. Therefore two values are determined for each cluster j: the cardinality of cluster j and the distance of j's centroid to the nearest macrocell. These values are used to create two ordered sets of clusters. Next, n_j is used to denote the position of cluster j in the set ordered by j's cardinality and d_j to denote the position of j in the set ordered by the distance of j's centroid to the nearest macrocell. Finally, a weighted sum of these two scores is generated as $v_j = \alpha n_j + (1-\alpha) d_j$, where $0 <= \alpha <= 1$ is the weighting factor. Once v_j is calculated, the

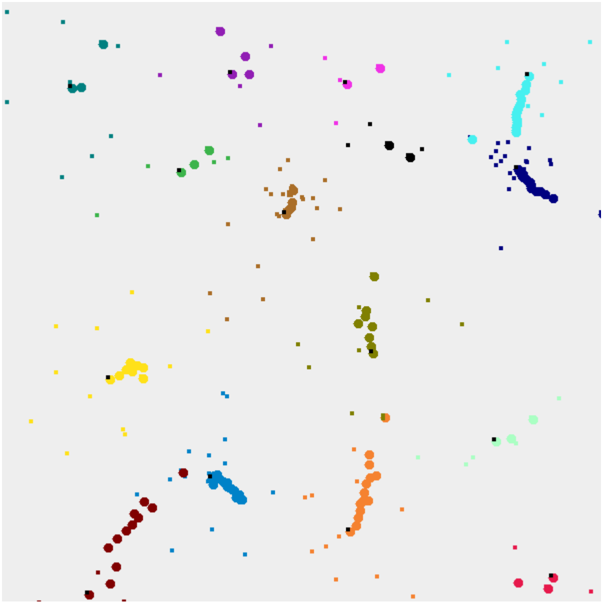


Figure 2.18: Evolution of K-Means++ Algorithm.

clusters are ordered by this quantity and the first D cluster's centroids are selected to be the coordinates of the drones. The result of this selection process is depicted in Figure 2.19 where the clusters represented in black are ignored and don't become UAV coordinates. As expected, since in this simulation there was a BS in the right side of the scenario, the colored clusters, that become UAV coordinates, are the ones that have a bigger amount of users, and/or are furthest away from the base station.

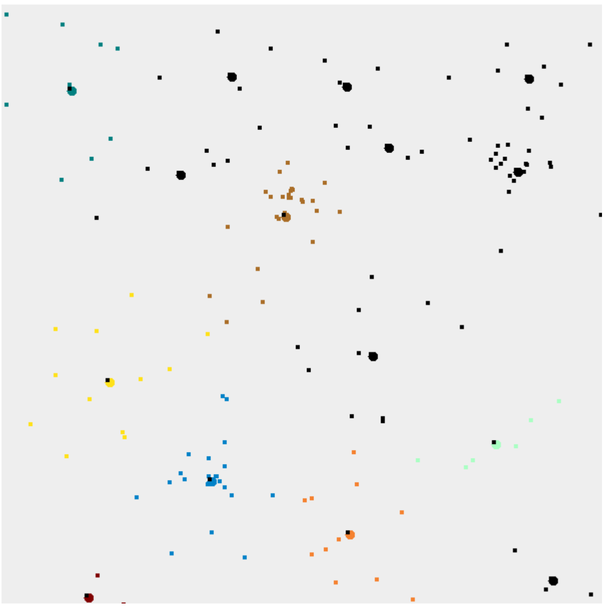


Figure 2.19: Cluster selection of K-Means++ Algorithm.

The time complexity of the K-Means algorithm is given by $\mathcal{O}(MNKI)$, where M is the number of points, N is the number of dimensions, K the number of clusters, and I the number of iterations.

Chapter 3

Optimization of Relay UAV Placement

3.1 System's Architecture

Figure 3.1 shows the general architecture of the proposed solution. The first thing to take into consideration is that, although there are two representations of the eNB, the functions they perform can be performed on the same eNB.

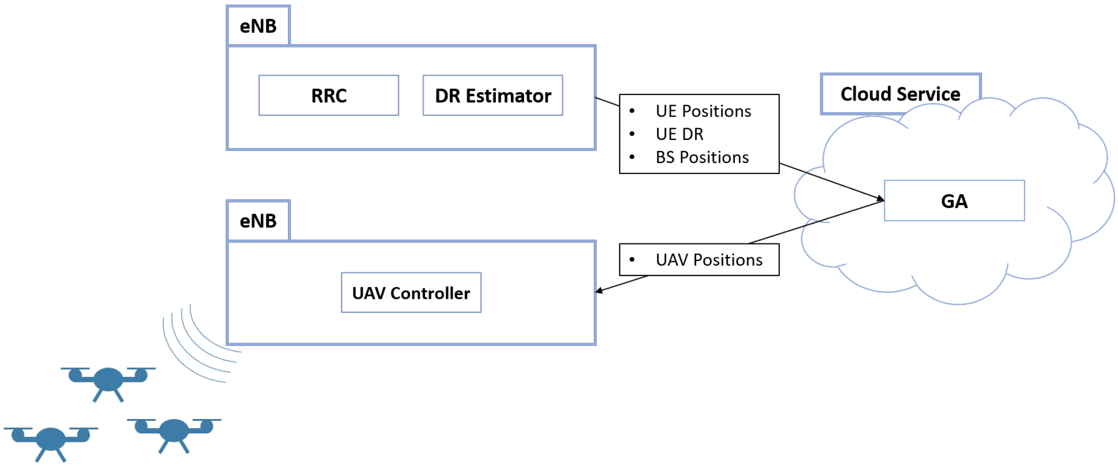


Figure 3.1: Architecture of the system.

In order to control the positioning of the UAVs the network would need an UAV Controller which would be located in the donor eNB. This element is the one that knows the UAV current coordinates, and, upon receiving the desired UAV Positions, can instruct the UAVs to move accordingly.

The decision about the UAV coordinates would take place in a cloud based service. This would allow the network operator to dynamically allocate resources to the application running the algorithm. The amount of processing resources the algorithm needs will depend on the amount of users being served at a current time, so, it is useful to dynamically change this configuration.

The GA is the algorithm chosen to perform the decisions about the UAV coordinates. In order to achieve this result, the algorithm needs three elements: the UE positions, the UE Data Rate (DR) requirements and the BS Positions.

In LTE, the resource allocation for each user works by increasing the number of RBs allocated to a UE until either the user is satisfied, or the scheduler maximizes the RBs to that user. Because of this policy, when a user requests service, the network does not know right away what is the DR that a UE needs. This is a metric that is constantly being adapted throughout time. In order to perform this estimation, a DR Estimator would be installed in the eNBs which would be constantly predicting the DR each user needs. The position of the UEs would also be estimated in the eNB, either by triangulation, or by receiving that information from the UE.

Finally, the UEs connect either to the assigned UAV or, if the UAVs run out of RBs, to the nearest BS. The RRC and DR Estimation for a certain user is located in its corresponding servicing eNB, and the UAV Controller is the the eNB that is coordinating the UAV operation.

Figure 3.2 shows the cycle that would guide the system. Beginning in the UE, this element starts by informing the network that it needs service and sends its coordinates. The DR Estimator, based on that request, and if applicable, on the previous requests from that UE, calculates or corrects the UE position and estimates the DR requirements of that UE. After the information about the UEs is complete, the DR Estimator sends that data to the cloud to be analyzed by the GA. The GA computes the new positions for the UAVs and sends them to the UAV Controller. The UAVs are then ordered to change their positions. With this reallocation, the radio conditions of the UEs change and the cycle begins again with the UEs either informing they don't need service anymore or asking for more resources.

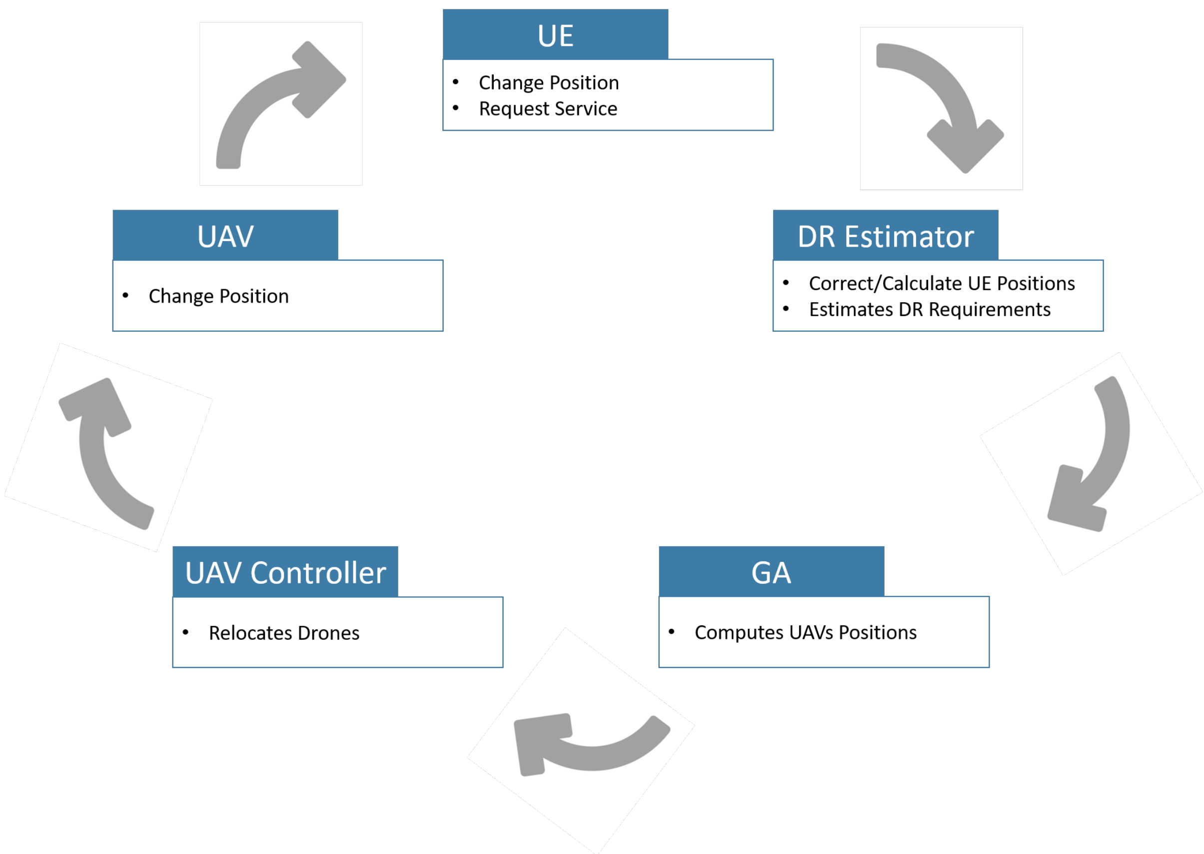


Figure 3.2: System Cycle.

3.2 Genetic Drone Disposition Algorithm

In the following subsections there will be a discussion of the implementation decisions relating the Genetic Algorithm.

3.2.1 Genetic Structure

As discussed in 2.3.1 the genetic algorithm is divided into 4 structural components: genes, chromosomes, individuals and generations. Figure 3.3 shows the logical bridge between the elements of the problem and the genetic elements.

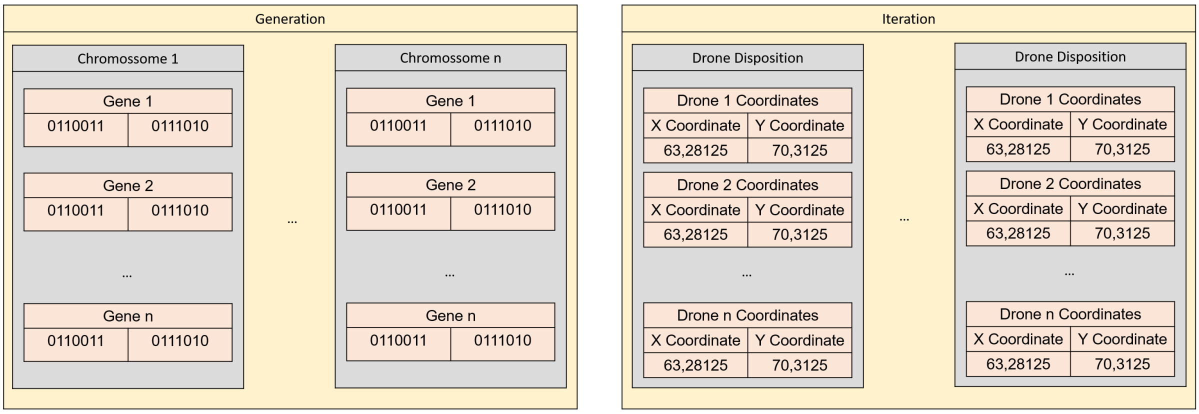


Figure 3.3: Genetic Structure Representation.

The genes should implement the most simple characteristic of the problem, which in our case is the coordinates of a single drone. The coordinates are represented in bits, and the implementation makes it possible to choose the level of granularity we want for our coordinates. This translates into changing the number of bits used to represent each coordinate. In Figure 3.4 there is an example of two different options for the coordinates representation. On the left, a two bits representation was chosen, and, on the right, a 3 bits representation was chosen. This means the yellow dot is represented by 1001 and the blue by 101100. Besides the obvious control of memory consumption, these different configurations allow a manipulation of the trade-off between the convergence time and the precision of the algorithm. In this case, the more bits are used, the more likely it is to get the perfect solution, since with less bits the perfect coordinates may not be possible to represent. However, since there are more options to go through, the more time consuming it becomes. In order to make the configuration more user friendly, the configuration file does not require the number of bits for every coordinate. Instead, it asks for the delta between two consecutive positions. From there, it calculates the number of bits necessary to fulfill that requirement. Requiring the delta instead of the number of bits, allows the user to be more sensible about what makes sense in the context of the telecommunications area.

The chromosome is the structure that holds the different genes. Every chromosome has the same number of genes, and since each gene encodes one drone, the number of genes will be equal to the number of drones. In our problem, each individual would have one and only one chromosome. This is why there is no formal distinction between these two structures.

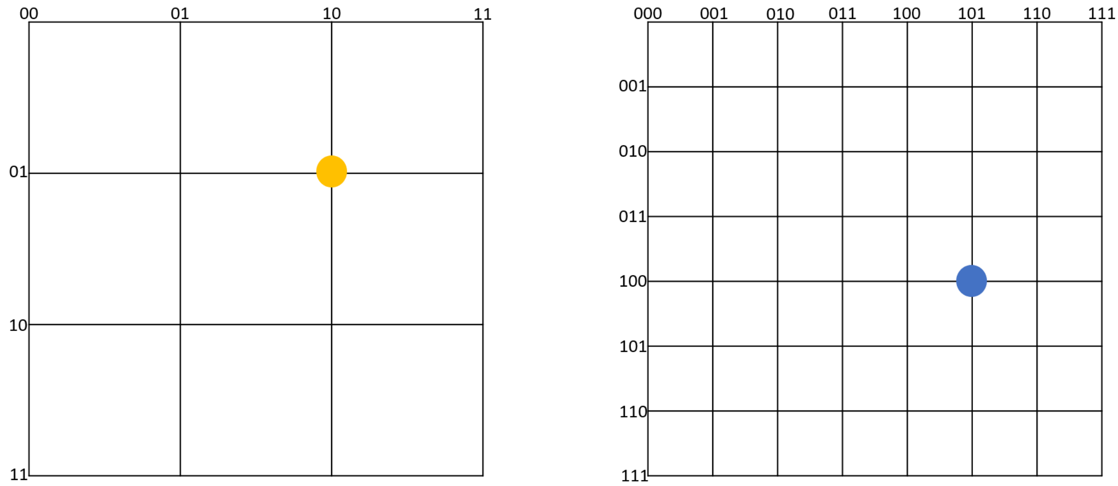


Figure 3.4: Coordinates representation with 2 bits on the left and 3 bits in the right.

The generation represents the population at any given moment and the number of individuals in each generation is also configurable.

3.2.2 Selection Implementation

The selection method implemented in this implementation is *dynamic*, because the selection probabilities are proportional to the actual fitness values, and not to the position of the individual in the fitness ranking. The selection is *preservative*, since every individual has a non-zero probability of being selected to reproduce, and is *elitist* because the top performing individuals (champions) are automatically selected to be in the next generation. The number of champions is an entry parameter.

As one can see from Figure 2.16, in order to iterate through generations it is necessary to recursively chose pairs of individuals to generate one descendent. This procedure starts by ranking all individuals by their fitness, then the algorithm chooses elements from the previous generation taking into consideration their fitness. They are randomly selected, although the probability of being chosen is directly proportional to their fitness values.

This selection is achieved by assigning to each individual a value of relative fitness:

$$f_i = \frac{F_i}{\sum_{i=0} F_i}, \tag{3.1}$$

where:

- F: absolute fitness value;
- f: relative fitness value.

After the selection process, the two chromosomes are crossedover and generate descendants.

3.2.3 Fitness Functions

The fitness function is used by the GA to classify the individuals. The goal is to improve the overall number of serviced users, and maximize the DR they can obtain. To achieve this, the algorithm runs the Telco Module for every individual and then uses the results that this module produces to classify the UAV configurations. The Telco Module will be further explored in the next section.

Three different fitness functions are implemented:

- Served Users: classifies the configurations by the number of UEs being serviced at the same time;
- Total DR: that returns the sum of DR obtain by all UEs;
- Proportion: the sum of the proportion, obtained by each user, between the required DR and the obtained DR:

$$F_i = \sum \frac{DR_{obt}}{DR_{req}}, \tag{3.2}$$

  where:

  - DR_{obt}: Obtained Data Rate by a user;
  - DR_{req}: Required Data Rate by a user.

The goal of the Proportion fitness function is to compromise between the two first metrics. This will make the algorithm prefer a disposition where the UEs with low requirements get some DR and then the UEs with high requirements are satisfied, to a disposition where the UEs with high requirements are fully satisfied before the UEs with low requirements are assigned some service.

3.2.4 Crossover Implementation

The crossover is the operation that joins the two different chromosomes, one from each parent, in order to create the new individual. As illustrated in Figure 3.5, the algorithm randomly chooses a point in the gene sequence to perform the crossover. All the genes before this point will come from one parent and all of the following will be from the other parent. As described in Section 2.3.3, it would be possible to implement a more complex crossover process. However, since the number of genes in each chromosome will be kept relatively small, it was not necessary.

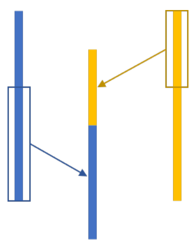


Figure 3.5: Crossover illustration.

There are two ways of implementing this operation. The first uses two parents to generate two descendants, and the second uses two parents to generate one descendant. This work implements the second option. Referencing Figure 3.5 again, in this example the algorithm chose the bottom of the blue chromosome and the top of the yellow chromosome to generate one child. The top of the blue chromosome and the bottom of the yellow are not used to generate another child.

3.2.5 Mutation Implementation

Mutation is an important contributor to environment adaptation. This is the mechanism that uses apparently random processes to create variability and, as previously discussed, variability is very important to generate adaptability. It is clear that mutation is important, however, it is not so obvious what is the optimum amount of mutation. In order to not hard code the amount of mutation in the algorithm, the amount of mutation is an input parameter. This parameter is divided into two sub-parameters: the *mutation rate* and *the mutation span*.

The mutation operation is applied right after a new individual is created, after the crossover. The *mutation rate* is translated in a probability of that new chromosome being mutated. The other important factor in mutation is how different should one let the mutated gene become. Preliminary experiments allowed to understand that if that quantity is too big, in our case, if we let the coordinates take any value, the algorithm tends to take much longer to converge to the final solution. This is why the algorithm implements a *mutation span*, illustrated in Figure 3.6, that only lets the coordinates change within a pre-defined range.

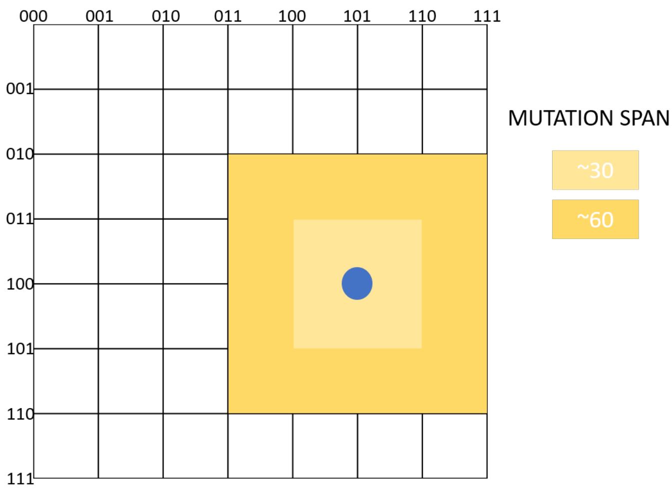


Figure 3.6: Mutation Span.

3.2.6 Implemented Stop Conditions

There are several ways to implement the stopping conditions for the algorithm. This work implements a stopping condition that counts the number of iterations performed, and when a pre-defined threshold is achieved the algorithm stops. One of the other options considered was stopping the algorithm taking into consideration the values of fitness of the previous generations. However, this methods could require adding complexity to the algorithm, and since the preliminary results indicated that the first option was sufficient, this was the chosen approach.

To arrive at this conclusion, several tests were performed on the genetic algorithm with multiple combinations of genetic parameters. In every test the conclusion was that it was possible to determine an iteration, beyond which any significant change was unlikely.

3.3 Telecom Model

Figure 3.7 shows the architecture of the algorithm that implements the telecom model. This algorithm starts by receiving the coordinates of all the UEs, all the UAVs and all BSs. Then it iterates through all UEs and assigns them to the available UAVs in order to allocate all of their capacity. The remaining UEs will be serviced by the BSs.

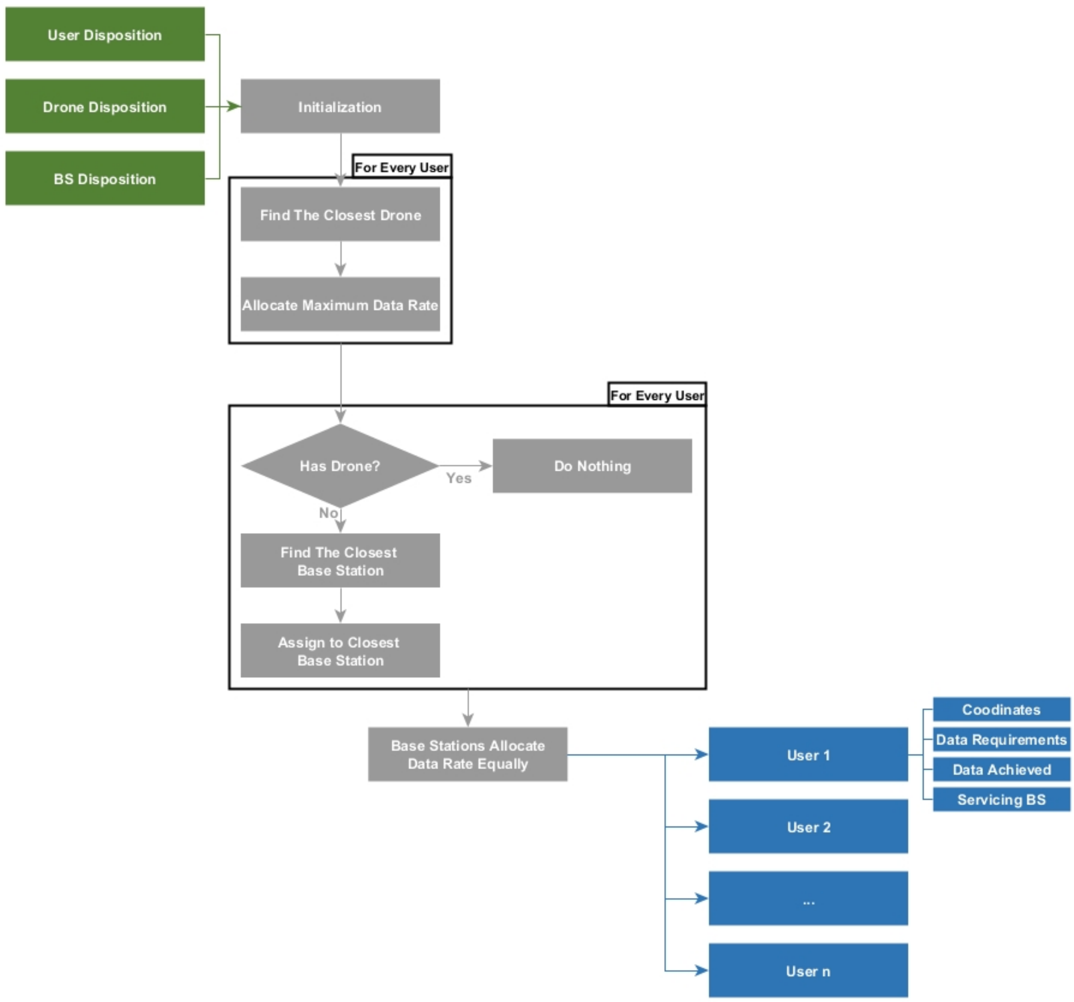
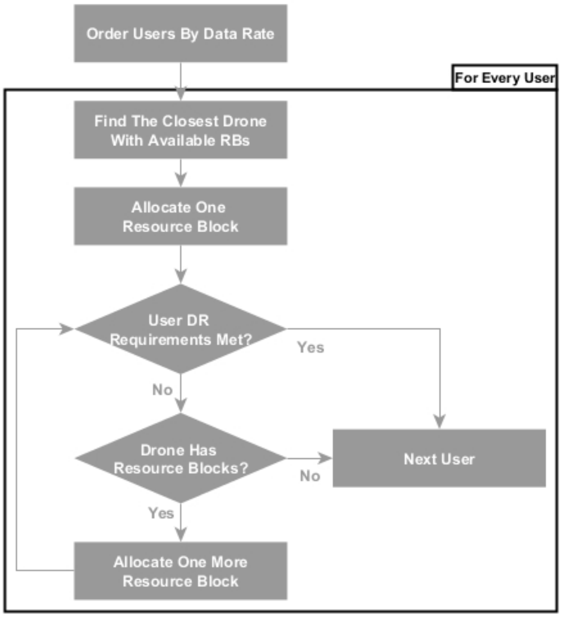


Figure 3.7: Telecom Model Architecture.

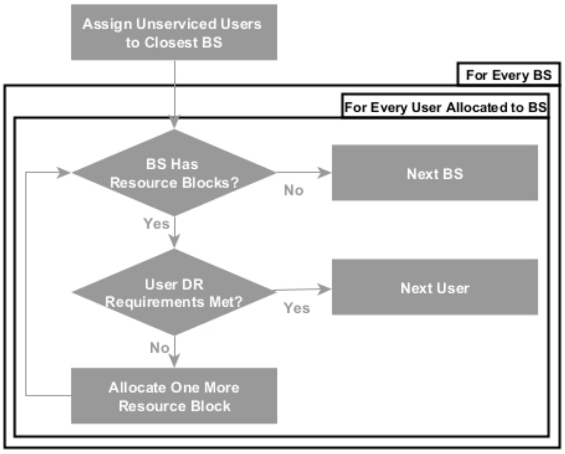
The algorithm implements two different schedulers for resource allocation, one for the allocation of resources in the UAV, and another for the resource allocation in the BS.

Figure 3.8 (a) shows the architecture of the UAV scheduler. This scheduler is a *Maximum Throughput Scheduler*. Since we want to maximize the UAV usage, the algorithm starts by ordering the UEs by data rate, serving the most greedy UEs first. For each UE, it will find the closest UAV with available resource blocks, and allocate RBs until either the UE data rate requirements are met, or the drone runs out of resource blocks.

Figure 3.8 (b) shows the architecture of the BS scheduler. This scheduler is a *Blind Equal Throughput Scheduler* which means it will implement a Round Robin Resource allocation, providing an equal number of RBs for every UE. It starts by assigning UEs to the closest BS, then it allocates one RB at a time to every UE until either the UE data rate requirements are met, or the BS runs out of resource blocks.



(a) Drone Scheduler



(b) Base Station Scheduler

Figure 3.8: Resource Allocation Schedulers.

It is important for both of these schedulers to make a prediction of how much data rate the UEs are going to obtain. There are mainly three links that need to be addressed for this porpuse. The first one is the link that will offload the UAVs (Backhaul Link), this link will be in LoS, and will be assumed to have enough capacity to offload the UAV necessities. Then, the Direct Link will either be provided in LoS by an UAV, or, in most cases, by NLoS by the BS. The following subsection will discuss both implementations.

Propagation Models

The goal of this sub-section is to describe how one can arrive at the data rate a UE obtains. This subsection is based in [6] and is implemented in the Telecommunications Module. To arrive at this value one needs to calculate some quantities, the first one is the available power at the receiver, which can be obtained as:

$$P_{r[dBm]} = P_{t[dBm]} + G_{t[dBi]} + G_{r[dBi]} - L_{p[dB]}, \tag{3.3}$$

where:

- $P_{t[dBm]}$: transmit power;
- $G_{t[dBi]}$: gain of the transmitting antenna;
- $G_{r[dBi]}$: gain of the receiving antenna;
- $L_{p[dB]}$: path loss.

The gains strongly depend on the type of the antenna. While for transmitting antennas the gain ranges from 15 to 21 dBi, the receiving antennas generally present gains from -5 dBi up to 10 dBi. The used gains in the simulator were respectively 15 dBi and 0 dBi. As for the transmit power, values range from 43 up to 48 dBm. We assumed BSs transmitted at 48 dBm and the UAVs transmitted at 43 dBm.

The path loss for the LoS links, considering the average power decay, is given by:

$$L_{p[dB]} = 32.44 + 20\ log(d_{[km]}) + 20\ log(f_{[MHz]}) + 10\ a_{pd}\ log(d_{km}), \tag{3.4}$$

where:

- $d_{[km]}$: distance;
- $f_{[MHz]}$: frequency;
- a_{pd}: average power decay, 2.

For the NLoS links the path loss can be calculated using the Okumura-Hata model, since the scenario presents itself as an urban or suburban. The deployment of drones intends to increase capacity, so it is expected that operators choose to use the frequency bands that give the most capacity, the 1800 MHz and the 2600 MHz. Since Okumura-Hata only models scenarios with frequencies ranging from 150 MHz to 2000 MHz, we opted to use the extension COST 231 to Hata's model which models scenarios with frequencies raging from 1.5 GHz to 2 GHz, which includes the two higher frequency bands. According to this model, the path loss can be calculated from:

$$\begin{aligned} L_{p[dB]} = {} & 46.30 + 33.90\ log(f_{[MHz]}) - 13.82\ log(h_{be[m]}) \\ & + [44.9 - 6.55\ log(h_{be[m]})]\ log(d_{[Km]}) \\ & - H_{mu[dB]} + C_{m[dB]} \\ & - \sum correction factors, \end{aligned}$$

where $h_{be[m]}$ is the effective height of BS antenna;

$$C_{m[dB]} = \begin{cases} 0, & smallcity; \\ 3, & urbancentres; \end{cases} \tag{3.5}$$

$$H_{mu[dB]} = \begin{cases} [1.10\ log(f_{[MHz]}) - 0.7]h_{m[m]} - [1.56\ log(f_{[MHz]}) - 0.8], & smallcitty; \\ 8.29\ log^2(1.54\ h_{m[m]}) - 1.0, & f \leq 200MHz, large\ city; \\ 3.20\ log^2(11.75\ h_{m[m]}) - 4.97, & f \geq 400MHz, large\ city; \end{cases} \quad (3.6)$$

where $h_{m[m]}$ is the UE's height.

To apply this model to an area that is somehow open spaced, we applied the correction factor for quasi open areas K_{op}, which for our transmitting frequencies yields 23 dB.

Link Capacity

An estimation of the data rate obtain by the UE can be obtained by:

$$r_b = B_{av[kHz]} \times R_{b,max[bit/s/Hz]}, \quad (3.7)$$

where:

- $B_{av[kHz]}$: is the available bandwidth for a specific UE;
- $R_{b,max[bit/s/Hz]}$: maximum data rate per hertz.

As mentioned in the subsection 2.1.6 the R_b, max can be calculated by 2.2, and the B_{av} is a result of the amount of resource blocks allocated to the UE. Manipulating the expression 2.1 we arrive at the UE's available bandwidth given by:

$$B_{av[kHz]} = N_{RB} \times B_{RB[kHz]} \times \frac{100}{P_{Bch[\%]}}, \quad (3.8)$$

where $P_{Bch[\%]}$ is the maximum data rate per hertz.

The Signal to Noise Ratio (ρ_N), is a function of the signal power received and the noise and interference:

$$\rho_N = \frac{P_{r[W]}}{N_{[W]} + I_{[W]}}, \quad (3.9)$$

where:

- $P_{r[w]}$: power received;
- $N_{[W]}$: thermal noise;
- $I_{[W]}$: interference.

We will assume a very low interference in comparison with the thermal noise due to the LoS and an efficient management of radio resources.

Finally, the thermal noise is given by:

$$N_{[W]} = K_{[m^2\ kg\ s^{-2}\ K^{-1}]} T_{[K]} B_{av[Hz]}, \tag{3.10}$$

where:

- $K_{[m^2\ kg\ s^{-2}\ K^{-1}]}$: boltzmann constant, $1.38064852x10^{-23}\ m^2\ kg\ s^{-2}\ K^{-1}$;
- $T_{[K]}$: temperature = 25.

3.4 Simulator

Figure 3.9 illustrates the general architecture of the developed simulator which was completely developed in Java. The blocks in blue are algorithms and will be addressed individually in the following sections. Initially, the simulator creates the UE disposition inside the scenario dimensions. The scenario is a rectangle with customizable dimensions, and the first task of the simulator is to generate positions and data rate requirements for UEs; this creates a UE disposition. Once that is done, the UE disposition is used by two algorithms which output a drone disposition each. The end goal is to compare the performance of these two drone dispositions.

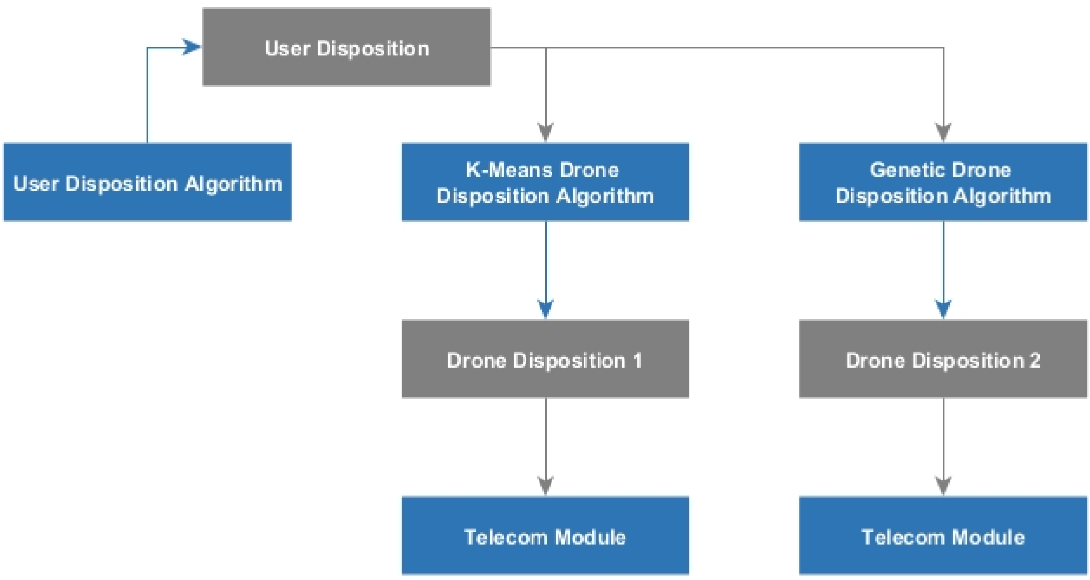


Figure 3.9: Simulator's architecture.

In order to compare the two drone dispositions, the simulator calls the Telco Module that receives the position of the drones and calculates the quality of signal received by each UE. With these parameters is possible to estimate the resulting data rate for each UE.

3.5 Simulator's General Components

The simulator's scenario is composed of a rectangle with customizable dimensions. This is the area where UEs are deployed. A UE has a set of coordinates and a data rate requirement, all UEs are assumed to be consuming network resources. In a real situation, not all UEs are using the network at the same time, so, from the simulator's point of view, it is as if those UEs don't exist. The drones will also be deployed inside this window. It is also possible to include a representation of one or multiple Base Stations (BS)/macrocells, however, the coordinates of these elements may be outside this window to include the possibility of the existing BSs being farther away. These BSs will serve the users that cannot be served by the UAVs.

The simulator assumes the existence of fixed based stations that will provide the Backhaul link for the UAVs. Both the BSs and the relay UAVs are assumed to be able to use the full bandwidth assigned for the LTE's operation. The relays are Outband and Full Duplex since they can communicate with the UEs and the Donor BS at the same time.

This work intends to make use of the different data rate requirements of UEs to more efficiently deploy the relay drones. The Table 2.5 shows the expected data rates for the different services the network may provide. We have decided to implement three categories of data rates for UEs: low demand, medium demand and high demand, and they will respectively be represented by 32 kbps/s, 200 kbps/s and 5 Mbps/s.

3.6 UE Disposition Algorithm

Representing the UEs uniformly distributed in the ground would be unrealistic since in reality there are some areas that have a bigger density of people than others. Taking that into consideration, the simulator uses a Poisson distribution to generate the UE's positions. It assumes some UEs will be agglomerated in small groups, or hotspots, and then distributes the remaining UEs uniformly throughout the scenario. Figure 3.10 depicts the architecture of this process.

This algorithm starts by randomly generating coordinates for the hotspots and UE coordinates within the hotspot radius. Once all hotspots are generated, the remaining UEs are generated uniformly in the scenario. As one can see in Figure 3.11, the algorithm receives as arguments the total number of UEs, the number of hotspots, the radius of the hotspots [m] and the average number of UEs per hotspot.

The number of UEs in each hotspot will be the result of a Poisson distribution with the average provided as an input argument.

For the distribution of DR requirements, two possible implementations were considered: Uniformly Distributed and Non-Uniformly Distributed. In the first case, the three possible levels for DR are distributed uniformly throughout the scenario.

In the second case, a random point is chosen from one of the corners of the scenario, then three circles with different radii are drawn around that point, as it is depicted in Figure 3.12 (a). These will represent areas of a certain type of data rate requirement, one for each of the three possible values.

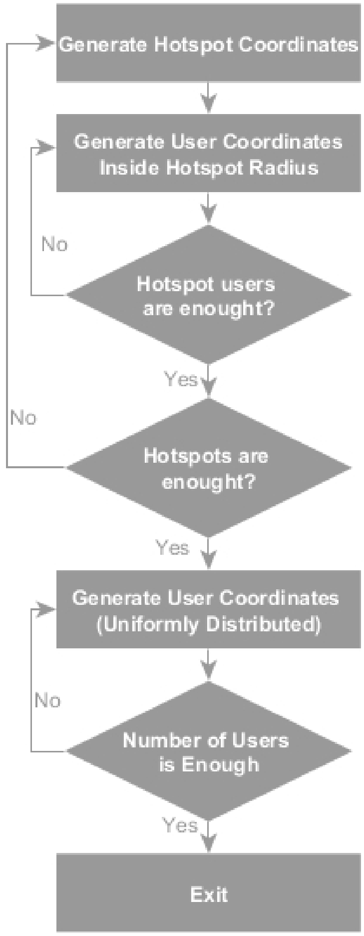


Figure 3.10: Poisson UE Disposition Algorithm Architecture.

When a user falls in one of those areas, it will be more probably assigned the respective data rate. The radiuses of these circles are chosen in a way so that every colored area has the same area. If a UE is inside one of those areas it has 50% probability of getting the level corresponding to that area and 50% probability of getting a random level of DR. If a user is not inside any the colored areas, it has equal probability of getting one of the three levels.

The Non-Uniform User Disposition was implemented to more realistically represent data rate requirements throughout the scenario. The assumption is that there are areas in the space that constitute hotspots, where the UEs require higher data rate. This can happen, for example, because there is some event happening in a specific place, or because there are groups of people more prone to require large amounts of data rate.

The output of this process is a list of UEs, each containing two coordinates and a data rate requirement in bit/s. Figure 3.12 (b) shows an example of an output from the UE Disposition Algorithm Non-Uniformly Distributed. This configuration has 300 UEs, 5 hotspots and, on average, each hotspot has 20 UEs. The color of the points represents the different data rate requirements of each UE: red UEs have the higher data rate requirements, the green UEs have the lower data rate requirements and the yellow UEs have a medium data rate requirements.

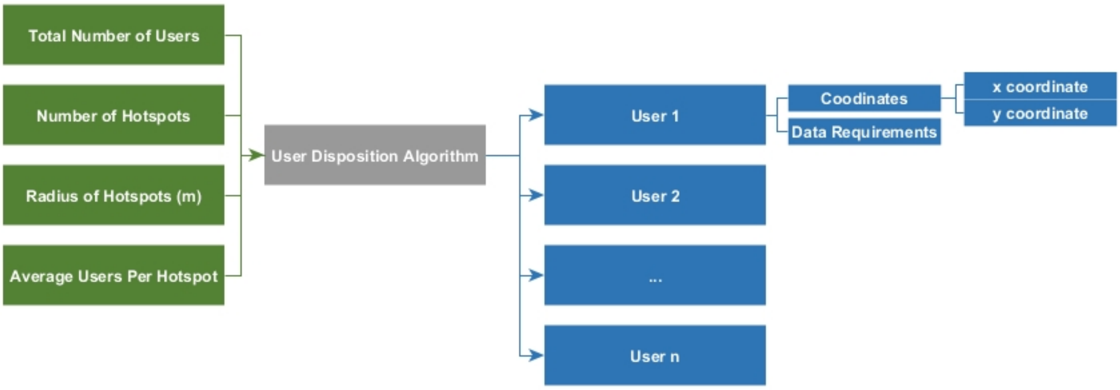
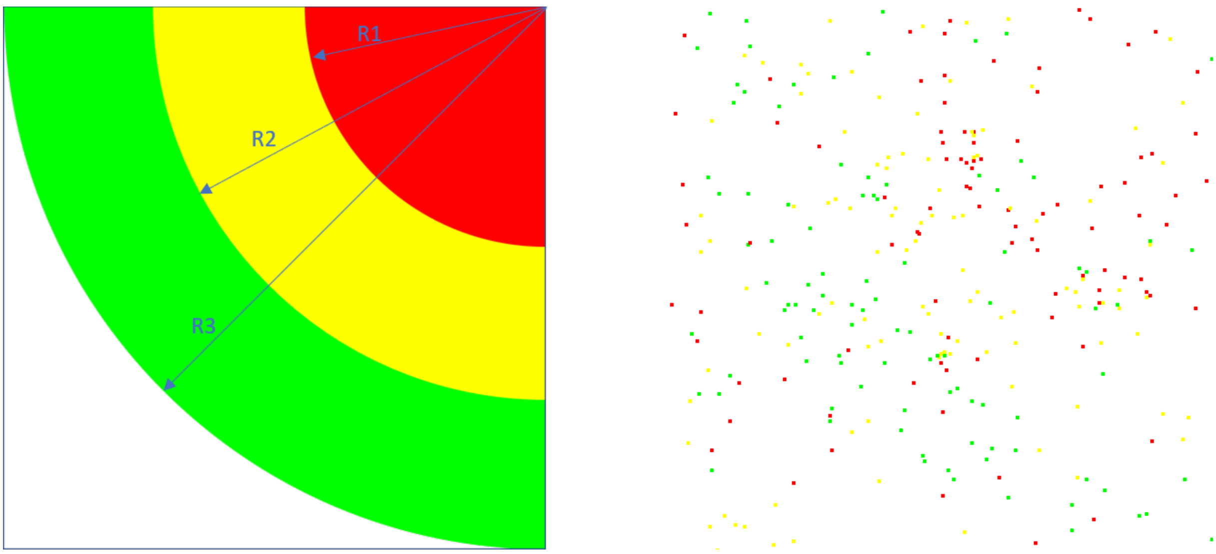


Figure 3.11: Poisson UE Disposition Arguments and Outputs.



(a) Data Rate Areas

(b) User Disposition Example

Figure 3.12: User Disposition.

Chapter 4

Simulation Results

This chapter presents the simulation results of this work. First, all the experiments relating the tuning of the GA will be shown, these experiments allowed to set the GA's parameters. Then, all the parameters for the scenario and user disposition are presented and justified. Finally, the results are shown and discussed.

4.1 Configurations

In order to properly deploy the simulations, the simulator requires the input of some specifications. The specifications are divided into categories and in the following tables (one for each category) there is the name, unit and description of every such configuration. In some cases, the value of the configuration is transversal to every simulation. In those cases, the used value is also depicted.

Scenario Configurations

Table 4.1 shows the possible scenario configurations. This allows for the configuration of the dimensions of the rectangular box where UEs and UAVs will be deployed. The only elements that can be outside this range are the BS/macrocells. The coordinates for these elements are defined in the Macrocell Configurations.

Table 4.1: Scenario Configurations.

Name	Unit	Description
X dimension	m	x maximum value for the scenario
Y dimension	m	y maximum value for the scenario

Macrocell Configurations

Table 4.2 shows the possible macrocell configurations. This configuration contains the coordinates of every macrocell. The transmit power can also be overrun here, but this parameter is not mandatory and is 48 dBm by default.

Table 4.2: Macrocell Configurations.

Name	Unit	Description
X coordinate	m	x coordinate of macrocell
Y coordinate	m	y coordinate of macrocell
Power	dBm	transmit power of the macrocell antenna

Drone Configurations

Table 4.3 shows the possible drone configurations. There is a drone capacity entry, however, it is important to stress that the drone capacity is calculated in run time depending on the users' data rate requirements. This variable is used to select the number of clusters to be calculated in the K-means++ algorithm.

Table 4.3: Drone Configurations.

Name	Unit	Description
Quantity	-	Number of available drones
Capacity	-	Number of users each drone can serve

User Disposition Configurations

Table 4.4 shows the possible user disposition configurations. These configurations are used for the Uniform User Disposition and the Non-Uniform User Disposition. The hotspots are areas where there will be a greater concentration of UEs and the size and concentration of UEs can defined here. The average parameter will be used by a Poisson Distribution to define the number of users in each hotspot.

Table 4.4: User Disposition Configurations.

Name	Unit	Description
Quantity	-	Total number of users
H quantity	-	Total number of hotspots
H radius	m	Hotspot's radius
Average	-	Average number of users per hotspot

Telco Configurations

Table 4.5 shows the possible configurations for the telco model. The values depicted for the *Frequency* and *Bandwidth* were selected according to the Table 2.3, and we decided to use one of the LTE's bandwidths that provide higher capacity, more specifically the 1800 band, because those bands are the more likely to be used in these kind of scenarios.

Table 4.5: Telco Configurations.

Name	Unit	Description	Value
Transmission Power	dBm	Antenna's transmission power	23
Transmission Gain	dBi	Transmission antenna's gain	15
Reception Gain	dBi	Reception antenna's gain	0
Frequency	kHz	Signal's central frequency	890 000
Bandwidth	kHz	Signal's bandwidth	20 000
Temperature	K	Temperature	303.15

GA Configurations

Table 4.6 shows the possible genetic algorithm configurations. These values were obtained via experimentation. The *Size* parameter was also determined experimentally. This should always yield a seven bit per coordinate value, but, since one can only know which size yields that number of bits when the size of the scenario is known, this parameter was left blank here.

Table 4.6: Genetic Configurations.

Name	Unit	Description	Value
Mutation Rate	%	Probability of gene mutation	30
Mutation Span	%	Possible variability on mutation	50
Population Size	-	Number of individuals in each generation	50
# Champions	-	Number of individuals to pass unchanged onto next generation	3
Size	m	Distance between two consecutive positions	-

4.2 GA Experimental Parameters

The results of a GA are dependent on the right configuration of its parameters. Since there is no rule to define these values, this section explains how they were obtained experimentally.

4.2.1 Evolution

Before performing any relevant simulation, some preliminary results were assessed. The first testes assessed the variation of the fitness from iteration to iteration. Figure 4.1 shows a representative example of the evolution of the GA.

These results are aligned with the theoretical expectations, since the fitness value increases quickly in the first generations, but then, as time passes, the algorithm converges to the optimum solution more slowly. As one can see, the algorithm reaches its final result around iteration 300. This number of iteration was used as reference for the following experiments.

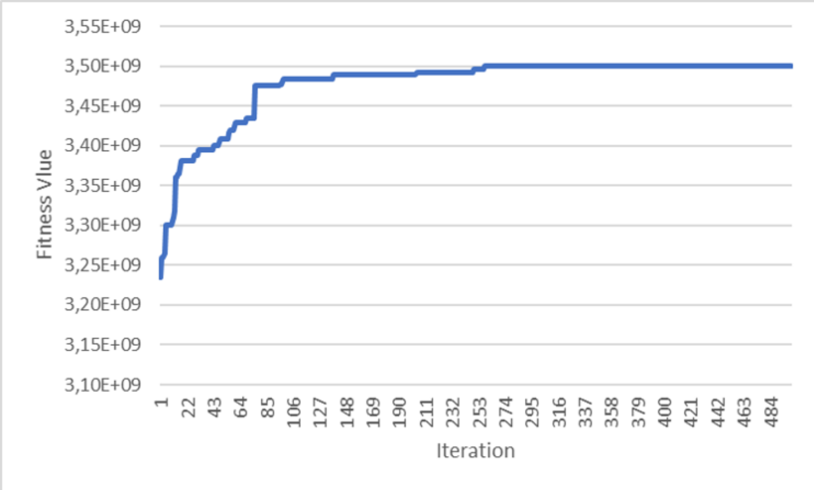


Figure 4.1: Fitness Value Variation over Time.

4.2.2 Mutation Rate

The first parameter to be defined was the *mutation rate*. Figure 4.2 shows the variation of the number of generations necessary to reach the final solution.

To obtain these values, the algorithm was run 20 times during 300 generations. It was assumed that at that point the algorithm had reached its final value. Then, the progress of the fitness value of each generation was evaluated and the first generation to achieve a fitness value different in no more than 1% than available in generation 300 was considered to be the first generation with the final value.

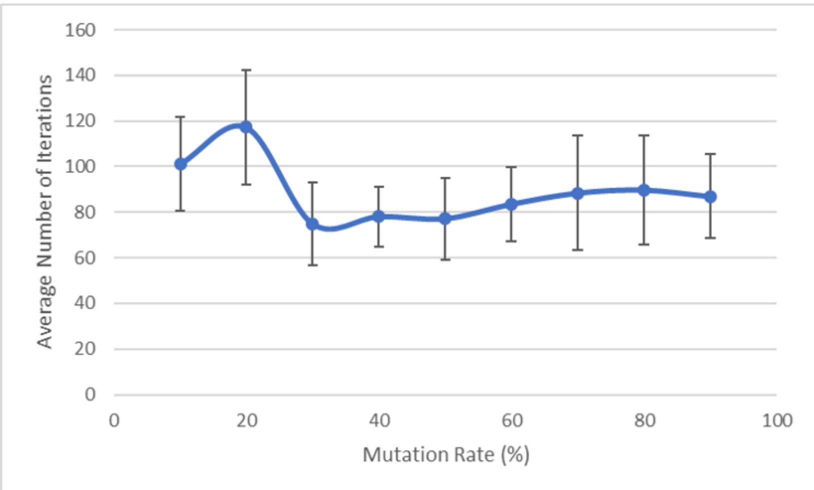


Figure 4.2: Convergence Speed Variation with the Mutation Rate.

As expected, the results indicate that a certain degree of mutation is useful to find the optimal solution faster. However, from a certain point onward, mutation has no effect, or has even a negative effect in the speed of convergence of the GA. This negative effect of high mutation rates can be explained by the algorithm not being allowed to properly converge to the right solution. Taking this results into consideration, the *mutation rate* was set to 30%.

4.2.3 Population Size

A similar experimental method to the mutation rate was made for determining the appropriate *population size*. Figure 4.3 shows the results of that experience.

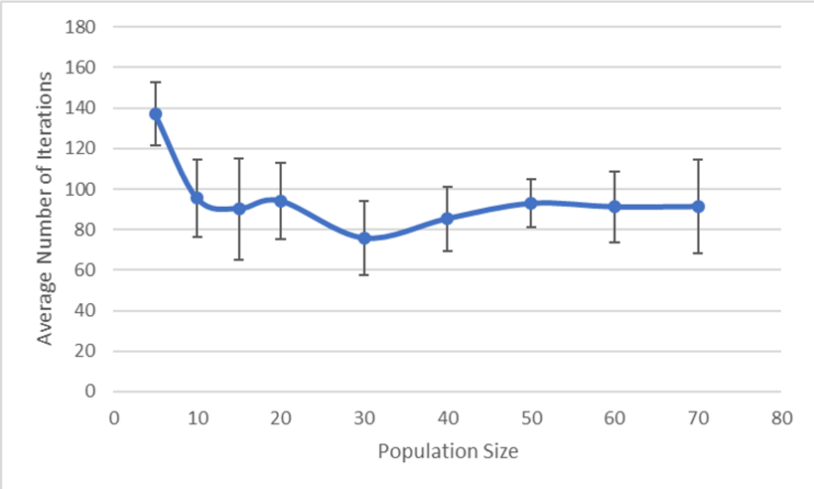


Figure 4.3: Convergence Speed Variation with the Population Size.

The results of this experiment reinforce the necessity of a minimum of *population size* to guarantee diversity. However, from a certain point onward, increasing the *population size* doesn't seem to affect the convergence speed of the algorithm. Taking these results into consideration, the *population size* was set to 50.

4.2.4 Number of Bits per Coordinate

The *number of bits per coordinate* determines the difference between two consecutive positions in the scenario. Table 4.7 shows this distance for different values of this parameter in a scenario with 300m x 300m. The smaller that distance, the more precise we expect the algorithm to become. In order to set this parameter correctly, two experiments were conducted. The first was similar to the ones performed for the *population size* and the *mutation rate*, which intends to study how fast the algorithm converges. The other experiment was to study the impact of this parameter in the correctness of the answer. For this last experiment, the algorithm was also run 20 times for 300 generations and the final fitness value was analyzed. Figure 4.4 and Figure 4.5 show the results of these tests.

The range of tested values ends in 7 bits per coordinate, because a distance smaller than 2 meters is not so relevant from the telecommunications perspective.

As expected, the algorithm converges at lower speeds with the increase of the *number of bits per coordinate*. This effect can be explained with the increase in possible position values for the drones's coordinates, making it less likely for the best drone configuration to appear early on.

The values of algorithm accuracy also increase with the increase of the *number of bits per coordinate*. These results confirm the expectations, with increases in the order of 2% for every extra bit.

Taking both tests into consideration, the *number of bits per coordinate* was set to 7.

Table 4.7: Distance between consecutive coordinates for different values of number of bits per coordinate.

Number of Bits	Distance (m)
2	75
3	37.5
4	18.75
5	9.375
6	4.6875
7	2.34375

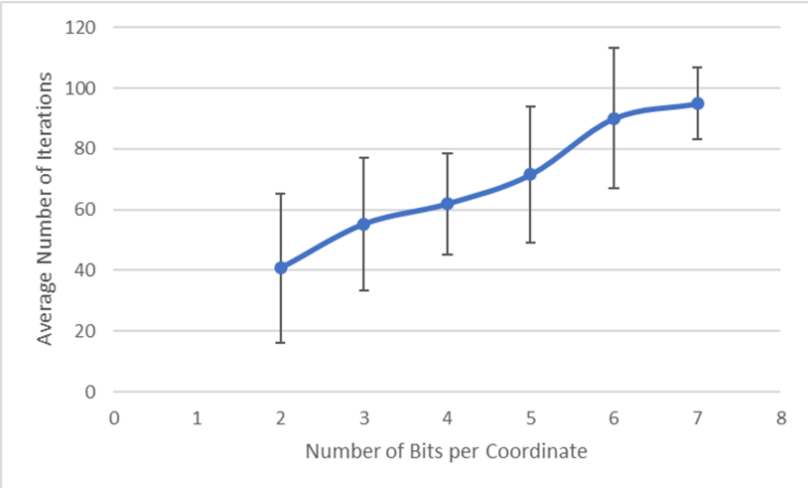


Figure 4.4: Convergence Speed Variation with the Number of Bits per Coordinate.

4.3 Simulations Description

In all the simulations both the GA and the K-Means++ were deployed. From this point on, the references to comparing the two algorithms, should be understood as the comparison between these two algorithms.

The scenario had a 300m x 300m dimensions, yielding a 90 000 m2 area. All simulations deployed two macrocell BS at the coordinates (150, 400) and (400, 150) and had 5 UAVs available.

Two performance metrics were evaluated:

- **Served DR**: this is the sum of all the DR obtained by each UE;
- **Served Users**: this is the sum of all UEs which were allocated at least one RB.

The first goal of the simulations was to understand the impact of the two possible UE's disposition regarding how DR is distributed throughout the UEs. To achieve that, both the Uniform and Non-Uniform Distributions were tested. The difference between these two configurations is depicted in 3.6.

The second goal of the simulations was to understand how different amounts of UEs affect the effectiveness of the two algorithms. To address that, two different amounts of UEs were deployed:

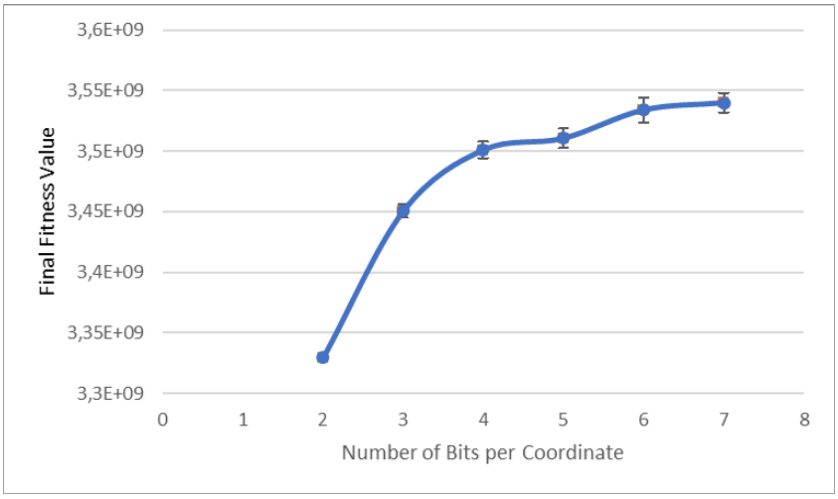


Figure 4.5: Final Fitness Value Variation with the Number of Bits per Coordinate.

- **Limit Case**: the number of UEs is expected to fully consume, or even marginally exceed the network resources;
- **Excess Case**: the number of UEs clearly exceeds the limit that the network is able to serve.

The limit case had 150 UEs in total, 5 hotspots, each hotspot had a 40 meters radius and on average 15 UEs. The Excess Case had 230 UEs, 5 hotspots, each hotspot had a 40 meters radius, and on average 20 UEs.

The third goal was to compare the performance of the two algorithms. The K-Means++ worked as a baseline. The GA has three possible fitness functions, as depicted in 3.2.3. The last goal of the simulations was to find the fitness function that yields the best results, and compare those results with the baseline.

Every combination of these possibilities was tested over 1000 Monte Carlo trials. Each trial followed these steps:

1. Create User Disposition
2. Run K-Means++ Algorithm
3. Run Genetic Algorithm
4. Run Telecom Module in both results

The results provided by the Telecom Module were analyzed, and the averages were computed with a level of significance of 95%.

4.4 Served Users Comparison

Table 4.8 and 4.9 show the results of the simulations for the number of served users.

Starting by analyzing the Non-Uniform scenario in the Limit case, all the fitness functions of the GA are good enough to assign service to all UEs, showing to be around 9% better than the K-Means++.

Table 4.8: Served Users by Algorithm in the Non-Uniform distribution.

			Served Users	
Scenario	**Algorithm**		***Average***	***C. L. (95%)***
Limit	**K-Means++**		136.91	2.4307
	GA	**Served Users**	150 (+8.73%)	0
		Served DR	150 (+8.73%)	0
		Proportion	150 (+8.73%)	0
Excess	**K-Means++**		225.4	0.5805
	GA	**Served Users**	230 (+3.07%)	0.3453
		Served DR	219.51 (-3.92%)	0.3453
		Proportion	229.98 (+3.06%)	0.0184

Table 4.9: Served Users by Algorithm in the Uniform distribution.

			Served Users	
Scenario	**Algorithm**		***Average***	***C. L. (95%)***
Limit	**K-Means++**		141,56	1,1932
	GA	**Served Users**	150 (+5.63%)	0
		Served DR	150 (+5.63%)	0
		Proportion	150 (+5.63%)	0
Excess	**K-Means++**		215,59	0,6630
	GA	**Served Users**	229,98 (+6.26%)	0,0159
		Served DR	210,64 (-2.15%)	0,5654
		Proportion	229,9 (+6.22%)	0,0394

In the Excess case, the AG was not always better than the K-Means++. In fact, it is possible to see that trying to maximize the global data rate leads to a starvation of some users since there are UEs who don't get any service assigned. However, for this metric, in this scenario, the other two fitness functions are very close to one another and around 3% or 6% better than K-Means++, depending on the way users are distributed in the scenario.

Comparing the Uniform and Non-Uniform distributions, it is obvious that in the Uniform scenario, the GA is not much better than the K-Means. The remaining conclusions are similar for both distributions.

4.5 Data Rate Comparison

The Table 4.10 and 4.11 show the results obtained for the served data rate. This value is obtained by summing the obtained data rate from all users.

When one looks at the Non-Uniform distribution results, the first thing to take notice is that, when the GA is using the Served Users fitness function, the results of this metric are very similar to the

Table 4.10: Served Data Rate by Algorithm in the Non-Uniform distribution.

			Served DR [bit/s]		Served DR [%]	
Scenario	**Algorithm**		***Average***	***C. L. (95%)***	***Average***	***C. L. (95%)***
Limit	**K-Means++**		2427205440	33531035.93	73.96	1.0216
	GA	**Served Users**	2437485884	30176476.36	74.27 (+0.31%)	0.9195
		Served DR	3238186945	2809427.52	98.67 (+24.71%)	0.0856
		Proportion	3216277558	3121521.93	97.99 (+24.04%)	0.0951
Excess	**K-Means++**		2808253150	23788282.68	50.92	0.4313
	GA	**Served Users**	2856064736	4439842.48	51.79 (+0.87%)	0.0805
		Served DR	3560569954	4439842.46	64.56 (+13.64%)	0,3074
		Proportion	3087416760	10004990.21	55.98 (+5.06%)	0.1814

Table 4.11: Served Data Rate by Algorithm in the Uniform distribution.

			Served DR [bit/s]		Served DR [%]	
Scenario	**Algorithm**		***Average***	***C. L. (95%)***	***Average***	***C. L. (95%)***
Limit	**K-Means++**		2551410483	15326588,07	83,94	0,4605
	GA	**Served Users**	2363379351	17054811,48	77,80 (-6.14%)	0,5556
		Served DR	3035138097	13508925,32	99,08 (15.14%)	0,1185
		Proportion	3000172608	12790525,51	98,83 (14.88%)	0,1429
Excess	**K-Means++**		2883848357	15603552,54	61,83	0,3739
	GA	**Served Users**	2830697170	10441104,35	60,71 (-1.11%)	0,2994
		Served DR	3411952986	3451926,188	73,27 (11.45%)	0,2682
		Proportion	3017640540	5678871,112	64,97 (+3.14%)	0,2403

K-Means++. However, the other two configurations of the GA outperform the K-Means by more than 24% in the Limit Scenario and by 5% or 13% in the Excess Scenario, depending on the chosen fitness function.

Just like in the Served Users Metric, when the algorithms are applied on the Uniform scenario, the difference between the two algorithms becomes less relevant. In some cases, for this UE disposition, the GA is even worse than the K-Means. However, it is always possible to find two fitness functions that yield better results than the K-Means.

As a final remark, it is visible that going from the Limit to the Excess Scenario, the amount of unserviced data rate becomes more and more significant, reducing the gap between the two algorithms.

4.6 Results Analysis

Regarding the results about GA parameterization, the experiments that were made show that the algorithm generally behaves as expected with the change of parameters.

The variations performed on the *mutation rate* indicate the correct implementation of the trade off between the convergence speed/global optimum discovery. This is specially obvious for low values of *mutation rate* for which the algorithm seems to be taking more time to converge to the local optimum due to a lack of variety in the population pool. However, it would be to expect that the algorithm also took more time to converge with high levels of mutation, since it would then not be able to converge.

The results for the variation of the *population size* on the convergence speed are also aligned with the theoretical expectations, since low values of *population size* do not seem to guarantee sufficient variability to rapidly find the global optimal solution. This increase in speed of convergence for this parameter is obtained by trying more options in every iteration.

The number of bits per coordinate also impacts the correctness of the solution as one would expect. With more bits per coordinate comes more precision in the determination of the coordinates, and thus, a more correct solution for the problem. This increase in precision also means there will be more options to be explored, and so, it is with no surprise that the time the algorithm takes to converge to the final solution also increases.

These experiments, aside from allowing to determine the best parameters, also let us confirm the correctness of the functioning of the GA.

As for the comparison of the results between the K-Means++ and the Genetic Algorithm, these results show that the GA has a better performance for almost every scenario. However, there is an important trade-off to consider, which is the balance between having a higher global data rate being consumed, versus having every user with access to the telecommunications service.

If one considers only the Limit Scenario, the fitness function that yields the best results is the Served DR, which maximizes the total data rate serviced by the network. This happens because the performance of the three fitness functions are similar in the number of serviced users metric and the Served DR fitness function is slightly better than the Proportion in the Served DR metric.

However, if one considers the Excess Scenario, the Served DR fitness function is no longer the obvious choice. Now the trade-off mentioned before is more evident. This fitness function yields now the worst results in the Served Users metric, worse than the K-Means++ algorithm. For this scenario, the wisest choice would be the Proportion fitness function, which compromises a little bit in Served DR, but keeps a high number of Served Users.

Taking this into consideration, the best fitness function would be the Proportion, since it improves the overall UE satisfaction, without compromising the number of served users. Table 4.12 summarizes the results for this fitness function.

These results show that in the best case scenario, the GA outperforms the K-Means by 24%. This is the case where the DR distribution is Non-Uniform, and the amount of data traffic is in the limits of the network capacity. When the DR is equally distributed, the K-Means improves its results and the difference is not so significant. Another relevant conclusion is that when the amount of unserviced DR increases too much (the Excess cases), the difference between the two algorithms also becomes less expressive.

The main conclusion of all these simulations is that if the fitness function is properly chosen, even in

Table 4.12: Summary of the most significant results.

UE Disposition	Scenario	Algorithm	Served Users	Served DR [%]
			Average	***Average***
Non-Uniform	**Limit**	**K-Means++**	136.91	73.96
		GA - Proportion	150 (+8.73%)	97.99 (+24.04%)
	Excess	**K-Means++**	225.4	51.79
		GA - Proportion	229.98 (+3.06%)	55.98 (+5.06%)
Uniform	**Limit**	**K-Means++**	141.56	83.96
		GA - Proportion	150 (+5.63%)	98.83 (+14.88%)
	Excess	**K-Means++**	215.59	61.83
		GA - Proportion	229.9 (+6.22%)	64.97 (+3.14%)

the worst case scenarios, the GA outperforms the K-Means.

Chapter 5

Conclusions

This thesis proposes a new solution for the optimization of telecommunications network patching using UAV relays. The related work was studied, analyzed and opportunities for improvement were identified. This proposed solution seeks to deploy UAVs to increase the capacity of the network meeting as much as possible the data rate requirements of the mobile users. The algorithm chosen to perform the optimization of UAV placement was a GA.

The various implementation options for these type of algorithm were analyzed and the tests performed to the functioning of the algorithm yielded expected results, showing that the implementation is functionally correct. Every configuration of the GA was analyzed and tested in order to tune this algorithm.

In order to evaluate the new approach, a baseline algorithm was also implemented. The chosen algorithm was the K-Means++. The preliminary tests of this algorithm also showed that the algorithm was performing as one would expect.

Two different user dispositions were used to compare the two algorithms. The GA showed to be better in both dispositions, in particular in the Non-Uniform disposition. Two different numbers of UEs were tested. The GA showed better results in both cases, while being particularly good in the Limit case, the case where the number of UEs is expected to fully consume, or even marginally exceed the network resources.

Two metrics were used to measure the quality of the algorithms: the number of Served Users and the Served DR. The GA proved to be able to not compromise the quality of any of them in detriment of the other. The GA was capable to keep the Served Users and still outperform the K-Means++ in the Served DR.

As for future work, it would be interesting to perform some extensions to this analysis. First, it would be useful to develop a better method to go from one state of UAV positions to another. For now the algorithm does not take into consideration the effort to change the positions of the UAVs. The GA could be changed to take into consideration this factor. This could result in lower quality UAV positioning, trading off with the amount of time the UAVs could stay in the air by reducing the effort needed to reallocate.

Another factor that was not fully addressed in this thesis is the frequency management necessary to deploy this solution. Some simplifications were made, specifically, assuming that every drone could make use of the full allocated spectrum. This is not exactly accurate because the drones may be relatively close to each other, causing interference if the frequencies used are not well chosen.

Thirdly, it would be interesting to deploy this solution in a more complex simulator to better evaluate the interactions between UAVs, UEs and eNBs. The previous point is a good example of a factor that a more complex simulator would not leave simplified.

It would also be interesting to evaluate the performance of other network indicators besides the number of served users and the average data rate per user. For example how would the delay be affected by a deployment of this type? And what would be the lifetime of UE's batteries when they are closer to the antennas?

Finally, it is important to better study the link between the UAVs and its serving BS. It is important to access the feasibility and trade-off of using the same frequencies as the ones used in the normal telecommunications operation, or if it would be better to use specific frequencies.

Bibliography

[1] W. OBILE, “Ericsson mobility report,” 2017.

[2] S. Sesia, M. Baker, and I. Toufik, *LTE-the UMTS long term evolution: from theory to practice*. John Wiley & Sons, 2011.

[3] C. Cox, *An introduction to LTE: LTE, LTE-advanced, SAE and 4G mobile communications*. John Wiley & Sons, 2014.

[4] T.-T. Tran, Y. Shin, and O.-S. Shin, “Overview of enabling technologies for 3gpp lte-advanced,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 54, 2012.

[5] H. Holma and A. Toskala, *LTE for UMTS: Evolution to LTE-advanced*. John Wiley & Sons, 2011.

[6] L.M.Correia, “Mobile communications systems – lecture notes,” instituto Superior Técnico, Lisbon, Portugal, 2017.

[7] “ANACOM anacom já definiu roteiro para a libertação e atribuição da faixa dos 700 mhz,” https://www.anacom.pt/render.jsp?contentId=1455674, accessed: 2018-06-29.

[8] “LTE Advanced Architecture lte advanced architecture — lte advanced protocol stack,” http://www.rfwireless-world.com/Terminology/LTE-Advanced-Architecture-and-protocol-stack.html, accessed: 2018-07-29.

[9] “OFDM mathematical description of ofdm,” http://www.wirelesscommunication.nl/reference/chaptr05/ofdm/ofdmmath.htm, accessed: 2018-06-26.

[10] S. Prasad, C. Shukla, and R. F. Chisab, “Performance analysis of ofdma in lte,” in *Computing Communication & Networking Technologies (ICCCNT), 2012 Third International Conference on*. IEEE, 2012, pp. 1–7.

[11] “ANACOM final report of the auction - final_report_auction.pdf,” https://www.anacom.pt/streaming/Final_Report_Auction.pdf?contentId=1115304&field=ATTACHED_FILE, accessed: 2018-06-29.

[12] “ANACOM microsoft word - decisão-decisão reshuffling final.docx - decisao_reshufflingmarco2012.pdf,” https://www.anacom.pt/streaming/Decisao_ReshufflingMarco2012.pdf?contentId=1120288&field=ATTACHED_FILE, accessed: 2018-06-29.

[13] O. Grøndalen, A. Zanella, K. Mahmood, M. Carpin, J. Rasool, and O. N. Østerbø, “Scheduling policies in time and frequency domains for lte downlink channel: a performance comparison,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 4, pp. 3345–3360, 2017.

[14] “Modulation optimizing rf signal performanceto improve lte coverage and capacity,” http://www.antennasonline.com/main/articles/optimizing-rf-signal-performance-to-improve-lte-coverage-and-capacity/, accessed: 2018-06-27.

[15] “Mobile technology trends mobile technology trends: traffic by application category,” https://www.ericsson.com/en/mobility-report/mobile-traffic-by-application-category, accessed: 2018-06-29.

[16] S. Pedrinho, “Análise do uso do lte-a mcc entre os militares em ambientes operacionais,” master Thesis, Instituto Superior Técnico, Lisbon, Portugal, 2017.

[17] R. Clarke, “Understanding the drone epidemic,” *Computer Law & Security Review*, vol. 30, no. 3, pp. 230–246, 2014.

[18] “Harvard Business Review 3 ways drones are changing telecom,” https://hbr.org/video/5418741099001/3-ways-drones-are-changing-telecom, accessed: 2018-06-26.

[19] C. F. Liew, D. DeLatte, N. Takeishi, and T. Yairi, “Recent developments in aerial robotics: An survey and prototypes overview,” *arXiv preprint arXiv:1711.10085*, 2017.

[20] S. Kandeepan, K. Gomez, L. Reynaud, and T. Rasheed, “Aerial-terrestrial communications: terrestrial cooperation and energy-efficient transmissions to aerial base stations,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 50, no. 4, pp. 2715–2735, 2014.

[21] R. C. Palat, A. Annamalau, and J. Reed, “Cooperative relaying for ad-hoc ground networks using swarm uavs,” in *Military Communications Conference, 2005. MILCOM 2005. IEEE*. IEEE, 2005, pp. 1588–1594.

[22] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, “Efficient 3-d placement of an aerial base station in next generation cellular networks,” in *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–5.

[23] A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal lap altitude for maximum coverage,” *IEEE Wireless Communications Letters*, vol. 3, no. 6, pp. 569–572, 2014.

[24] I. Bor-Yaliniz and H. Yanikomeroglu, “The new frontier in ran heterogeneity: Multi-tier drone-cells,” *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–55, 2016.

[25] B. Galkin, J. Kibilda, and L. A. DaSilva, “Deployment of uav-mounted access points according to spatial user locations in two-tier cellular networks,” in *Wireless Days (WD), 2016*. IEEE, 2016, pp. 1–6.

[26] M. Mitchell, *An introduction to genetic algorithms*. MIT press, 1998.

[27] B. L. Miller and D. E. Goldberg, “Genetic algorithms, selection schemes, and the varying effects of noise,” *Evolutionary computation*, vol. 4, no. 2, pp. 113–131, 1996.

[28] T. Bäck and F. Hoffmeister, “Extended selection mechanisms in genetic algorithms,” 1991.

[29] M. Safe, J. Carballido, I. Ponzoni, and N. Brignole, “On stopping criteria for genetic algorithms,” in *Brazilian Symposium on Artificial Intelligence*. Springer, 2004, pp. 405–413.

[30] J. A. Hartigan and M. A. Wong, “Algorithm as 136: A k-means clustering algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[31] D. Arthur and S. Vassilvitskii, “k-means++: The advantages of careful seeding,” in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.