4-2019

Aerial Base Station Deployment for Post-Disaster Public Safety Applications

Jose Antonio Matamoros Vargas
University of Nebraska-Lincoln, jose.matamoros-vargas@huskers.unl.edu

MOBILE AERIAL BASE STATION DEPLOYMENT FOR POST-DISASTER PUBLIC SAFETY APPLICATIONS

by

Jose Antonio Matamoros Vargas

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Telecommunications Engineering

Under the Supervision of Professor Yi Qian

Lincoln, Nebraska

April, 2019

MOBILE AERIAL BASE STATION DEPLOYMENT FOR POST-DISASTER PUBLIC SAFETY APPLICATIONS

Jose Antonio Matamoros Vargas, M.S.

University of Nebraska, 2019

Adviser: Yi Qian

Earthquakes and floods are constant threats to most of the countries in the world. After such catastrophes, a rapid response is needed, which includes communications not only for first responders but also for local civilians. Even though there are technologies and specialized personnel for rapid deployment, it is common that external factors will hinder the arrival of help while communication requirements are urgently required. Such communication technologies would aid tasks regarding organization and information dissemination from authorities to the civilians and vice-versa. This necessity is due to protocols and applications to allocate the number of emergency resources per location and to locate missing people.

In this thesis, we investigate the deployment problem of Mobile Aerial Base Stations (MABS). Our main objective is to ensure periodic wireless communication for geographically spread User Equipment (UE) based on LTE technology.

First, we establish a precedent of emergency situations where MABS would be useful. We also provide an introduction to the study and work conducted in this thesis.

Second, we provide a literature review of existing solutions was made to determine the advantages and disadvantages of certain technologies regarding the described necessity.

Third, we determine how MABS, such as gliders or light tactical balloons that are assumed to be moving at an average speed of 50 km/h, will be deployed.

These MABS would visit different cluster centroids determined by an Affinity Propagation Clustering algorithm. Additionally, a combination of graph theory and Genetic Algorithm (GA) is implemented through mutators and fitness functions to obtain best flyable paths through an evolution pool of 100.

Additionally, Poisson, Normal, and Uniform distributions are utilized to determine the amount of Base Stations and UEs. Then, for every distribution combination, a set of simulations is conducted to obtain the best flyable paths. Serviced UE performance indicators of algorithm efficiency are analyzed to determine whether the applied algorithm is effective in providing a solution to the presented problem.

Finally, in Chapter 5, we conclude our work by supporting that the proposed model would suffice the needs of mobile users given the proposed emergency scenario.

ACKNOWLEDGMENTS

I would first like to thank my thesis advisor, Dr. Yi Qian of the Electrical and Computing Engineering Department at at the University of Nebraska-Lincoln. The provided suggestions, recommendations, and aperture were highly helpful in order to correctly fulfill this thesis. Second I would like o thank Shengjie Xu, who has provided his guidance and recommendations for my research. His time and attention were essential for me in order to reach my thesis goal. Consequently I would like to thank the Computer Networks and Security (CNS) group, with whom I had frequent meetings to establish weekly updates and goals. The information and suggestions from our open discussions were always welcomed and helpful when it came to review the work being done.

Second, I would also like to thank my career advisor, Dr. Hamid Sharif of the Electrical and Computing Engineering Department at at the University of Nebraska-Lincoln. His guidance for my academic evolution is something I appreciate very much. His recommendations allowed me to take the utmost knowledgeable and advantageous courses during my studies. In addition, I appreciate the open doors and friendliness to help me with any problem I had.

Third, I would like to acknowledge the effort of my thesis committee for taking the time to review this thesis.

In addition, I would like to thank the Fulbright Organization. They provided me with the incredible opportunity to fulfill my Masters studies in the United States. Furthermore, I would like to show my appreciation to the University of Nebraska for providing an extraordinary experience regarding intercultural experiences within a highly academic environment.

Finally, I would like to express my gratitude to my parents and siblings, who have provided support for my current endeavors. To my friends, from whom I have gained knowledge and shared learning and joyful experiences.

This accomplishment would not have been possible without them. Thank you. This thesis was partially supported by the National Science Foundation under grant EARS-1547330.

Table of Contents

List of Figures

List of Tables

Chapter 1

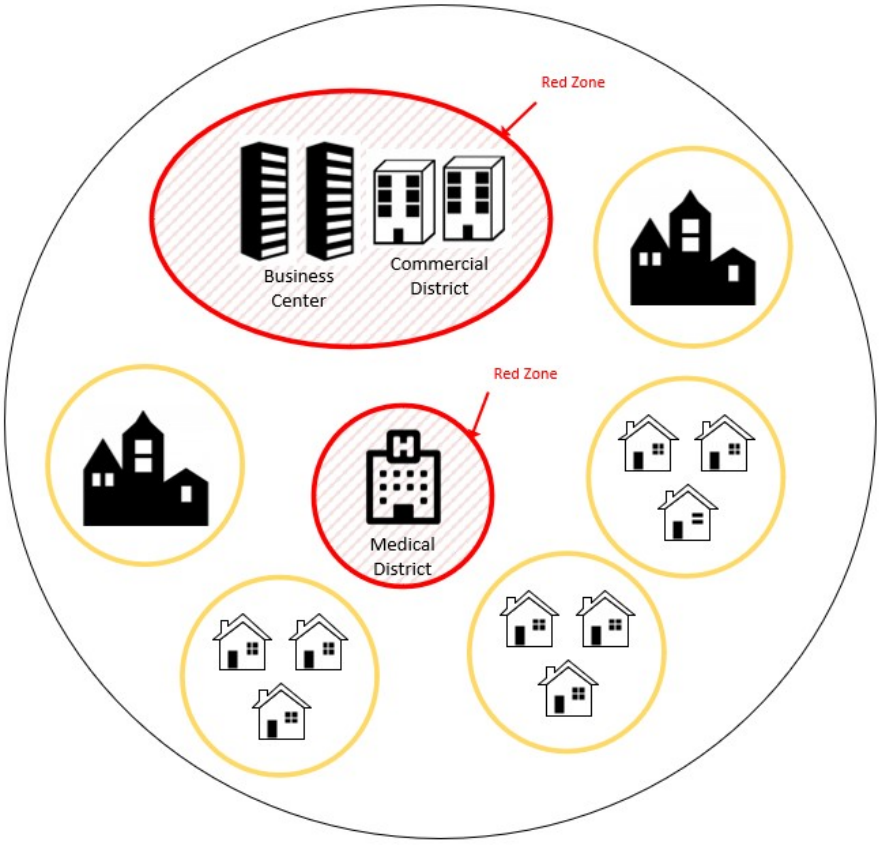
Introduction

On the course of history, many natural disasters caused by earthquakes, hurricanes, tsunamis, floods, tornadoes, and other hazards have threatened the daily life of human beings. Nowadays people's way of life heavily depends on deployed and maintained infrastructure that covers transportation, communications, food supply, and health services.

After a natural disaster in a small-to-mid populated area (i.e., that usually has a population between 100,000 to 1,000,000 inhabitants [3]), focus areas (Red Zones) for rescue missions will be promptly scheduled for a visit with the objective of saving as many lives as possible after a brief risk-infrastructure/situation analysis. Figure 1.1 shows an after-disaster scenario, different city locations are designated differently to achieve an utmost effective rescue mission. If we assume that the disaster was spontaneous (e.g., an earthquake) and during business hours, then a business center and commercial districts might be where most of the local citizens will be. Therefore, these zones, in addition to medical centers will be tagged as red zones due to the likeliness of a great number of people being at those sites. Moreover, this will render residential zones a different status, since the population density will be deemed lower at that moment.

In addition, Radio Access Points for telecommunications services may become unavailable due to infrastructure damage. This includes cellular towers,

Figure 1.1: Post-Disaster Red Zonification



in-home Wi-Fi installations, public Wi-Fi hotspots, and landline based technologies (e.g., ADSL, FTTH). The level of damage to such infrastructure solely depends on a case-by-case scenario, due to the nature of the hazardous event, and currently cannot be predicted with much accuracy. This will result in a scenario where all rescue resources (e.g., rescue personnel, emergency communication systems, emergency health services, and rescue specialized devices and machinery) will be directed to the Red Zones.

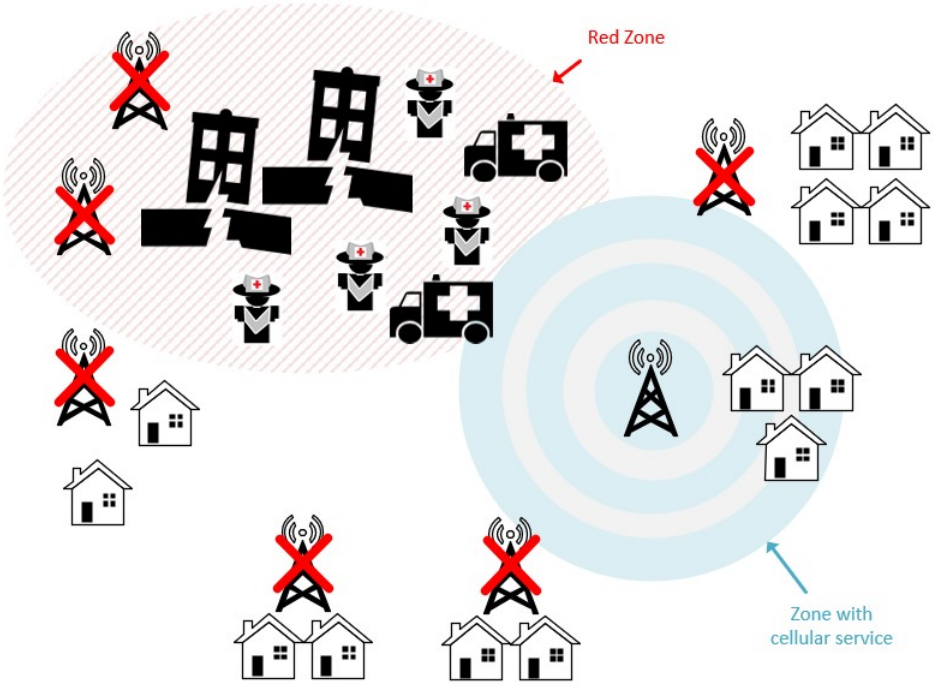
While it is imperative to focus resources on Red Zones, this does not mean that the rest of the surrounding population does not require communication services. In this case, even non-continuous access to an Internet Relay Chat (IRC) service will help uncommunicated civilians access information about their relatives and public notifications about the current public safety situation.

Those services will allow civilians to be informed and controlled during the aftermath of the disaster event. Such necessities are brought to attention by [4] [5] [6] [7] [8], where the authors analyze different post-disaster scenarios and their communication needs. It is assumed that most of the civilians will be carrying a device capable of communicating with a wireless network. These devices will be considered User Equipment (UE), and it will have the ability to establish communication with LTE Base Stations (BS).

The response handled by authorities to such catastrophes can be classified into a primary stage, an active stage, and a late-stage rescue service.

During the primary stage, we may consider a scenario where a help request is being sent, but aid arrives in different conditions and dates (e.g., where an emergency communication system is sent by parts, and the radios antennas and control center arrive 1 or 2 days later due to size and weight restrictions and limitations during transportation) depending on the nature and location of the disaster event. For example, in a developing country, after an earthquake, international help may arrive, but because of conditions out of rescuers jurisdiction (e.g., delay in customs, stalled airport services, military checkpoints, or being a remote location), it may be stalled or interrupted; forcing rescuers to work with the available resources. This stage usually lasts from two days to a week depending on unforeseen conditions [9]. An illustration to a primary stage after an earthquake can be found on Figure 1.2. Here we assume that wireline communication connections are unavailable, and distributed Base Stations have been affected by the seismic event. Because of the nature of the event, rescue teams will focus their attention to the now assigned Red Zone. In Figure 1.3 a similar situation can be found. After a severe flood, highly populated places are attended by rescuers before others. In both scenarios, we can appreciate how the lower populated areas will suddenly become partially uncommunicated until further help arrives (i.e., a communication solution is deployed).

Figure 1.2: Primary Stage for Earthquake Scenario



During the active stage, we may assume that all resources available are at hand, and effective deployments are done. Here, Cell-On-Wheels (COW), remote hospitals, remote control towers for air traffic, and refugees areas (with tents and access to basic needs), for example, may be adequately deployed, thus allowing public safety personnel to achieve their job effectively. This stage will usually last from 2 weeks to a month.

A different scenario comes to be when the situation arrives at an active stage. As seen in Figure 1.4 COWs have been deployed, and more rescue personnel have arrived at the affected sites. This allows a more effective rescue and re-organization scenario. While in Figure 1.5 we can appreciate how in the active stage, in addition to COWs, Aerial Vehicles have been directed to the different sites. This aerial vehicle may work as an Unmanned Aerial Base Stations (UABS), or as security video footage source for terrain condition evaluation. Overall, it is comprehensible how active stage is where the utmost adequate technology for disaster relief is deployed.

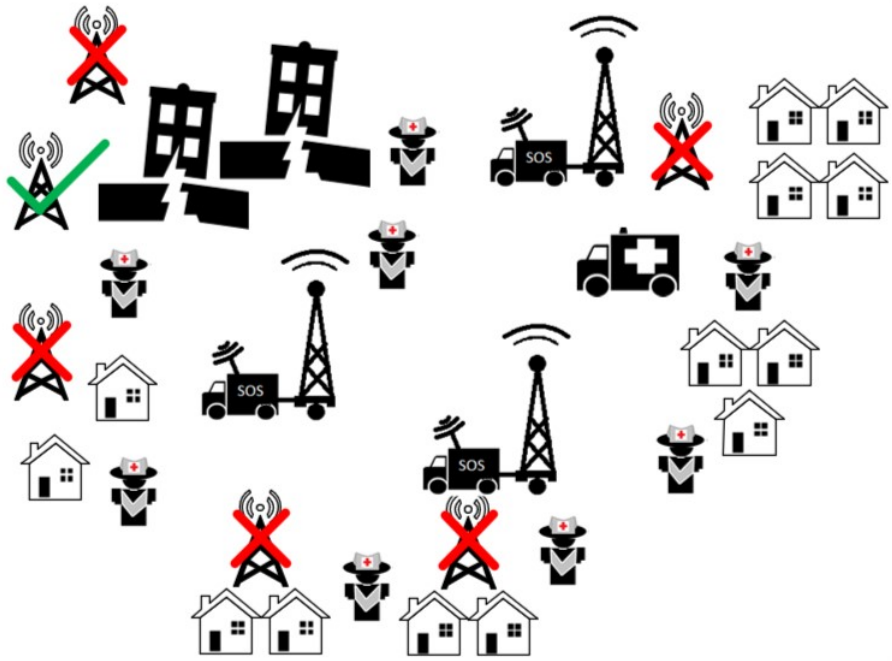
Figure 1.3: Primary Stage for Flood Scenario

During the late stage, some services, and equipment from the active stage usually are kept until the whole rescue and logistics operation has ended, and basic services are re-established. For example, this may include some COW and some refugee camps being left for longer periods. This stage lasts in accordance with a scenario basis.

This work will focus on the primary stage, where resources are scarce and better solutions may arrive later. Because of human nature, they usually spend one or two weeks near their homes to protect their personal belongings while not being bale to access their house for integrity/security reason (i.e., the house may suffer from structural damage). Or they may be asked to stay near their houses for public safety logistical operations (e.g., water and food distribution services, and unsafe traveling roads).

Meanwhile, Red Zones will have improved communication deployments, and

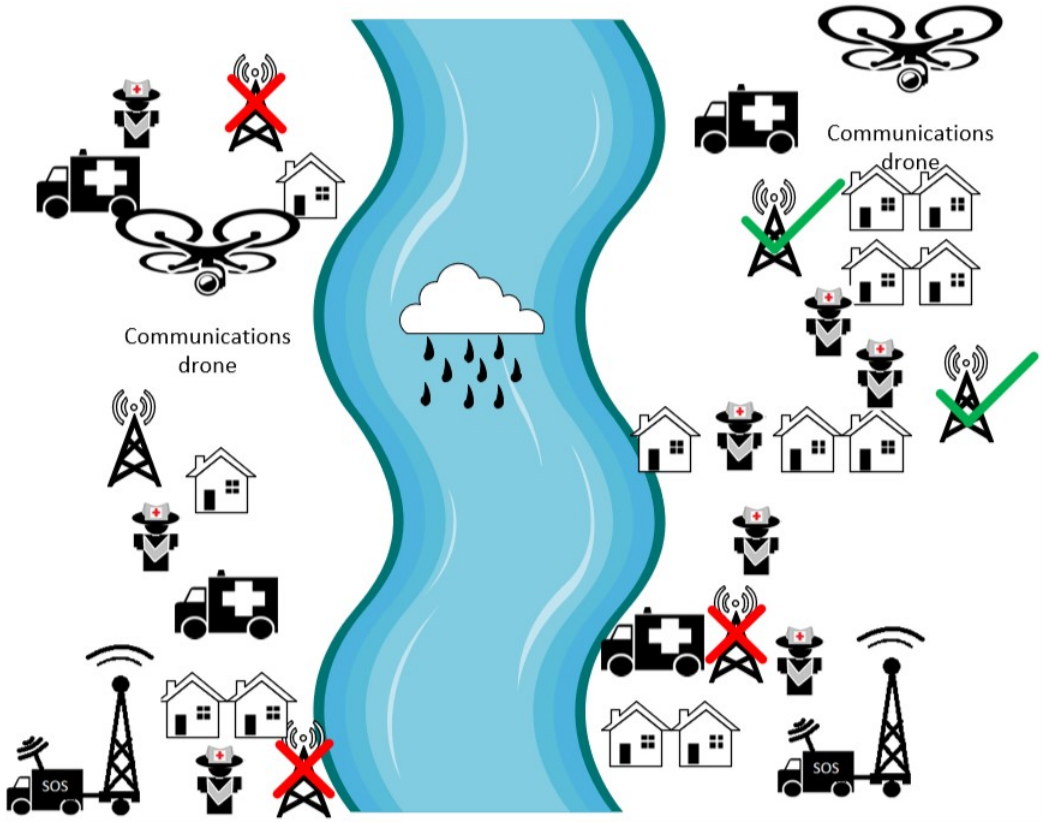
Figure 1.4: Active Stage for Earthquake Scenario



some zones may not suffer from a lack of communication after the seismic event. This means that during the first days (i.e., primary stage), UE outside of the Red Zone may be uncommunicated, and the only information source civilians outside the red zone may have could be a military patrol conveying messages through a loudspeaker. This is not an adequate communication infrastructure for people's needs to be informed about their relatives or provide information to the authorities about a missing person (information important for authorities while on rescue missions); tasks that can be solved by providing partial communications to UEs given the current scenario. To aid such need and to provide a better communication scenario for people outside Red Zones, solutions such as COW, data satellite enabled routers, low band wireless phone services implementations (e.g., CDMA-450), and satellite phones can be employed. However these usually are provided to the UE during the active or late stage.

In this chapter, we portrayed emergency situations where periodic communication would allow UE to access limited but important short messages

Figure 1.5: Active Stage for Flood Scenario



or information bulletins through an IRC.

In Chapter 2, we perform a survey of Disaster Recovery Networks (DRN) and their applications during emergency situations. We review how Land Mobile Radio (LMR) is used for emergency communications, and study successful application deployments in the United States of America. Satellite Communications (SatComs) are studied from an emergency deployment perspective and analyze the technological advancements achieved within this technology. Cell On Wheels (COW) are studied, and real-life deployments are analyzed in order to determine their emergency responsiveness reach. D2D communications for public safety scenarios are reviewed regarding their improvement and emergency aid reach. Unmanned Aerial Vehicles (UAV) for emergency deployment research is studied and analyzed given while considering our described situation. Lastly, this chapter is concluded by summarizing the

advantages and disadvantages of each surveyed technology.

In Chapter 3, a Mobile Aerial Base Station (MABS) path determining algorithm is proposed. First, the problem is presented as a Multiple Traveler Salesman Problem (MTSP). In this problem, sets of uncommunicated UE (i.e., UE with a channel bandwidth below 200 Kbps which would be insufficient for a simple IRC) are clustered by employing an Affinity Propagation Algorithm and determining centroids. Afterward, these centroids are applied to establish a graph $G = (V, E)$, where V are the centroids, and E are paths to be visited by MABS. To determine the paths, a joined graph theoretical approach and Genetic Algorithm is implemented. For this, mutation and fitness functions are implemented to establish the best solution to the given problem.

In Chapter 4, a series of simulations are conducted to determine the functionality of the proposed algorithm. To provide a closer approach to a real-life situation, Poisson, Normal, and Uniform distributions are simulated to determine the density of UE and BS for a combination of simulations. In addition, every combination will be simulated with a number of one to five allowed MABS. Lastly, we analyze the performance of the algorithm given the different simulated scenarios and conditions.

In Chapter 5, we provide a conclusion to our work and provide a briefing of future work to be investigated.

The major contribution of this thesis resides in the study of current emergency response technologies and a proposed algorithm to establish periodic communication for uncommunicated UE given a post-disaster situation during a primary stage with limited resources.

Chapter 2

Literature Review

Much work has been done in public safety communication solutions for disaster areas. For example, in [10][11] the authors analyze the advances in Public Safety Communication Systems and challenges in various fields. Different solutions are then proposed considering different emergency scenarios and assumptions. In this chapter, we analyze different aspects and approaches for emergency response networks and provide the research path that leads to our work.

In [12] [13] [14] [15] the authors consider network robustness for communication infrastructure. They explore opportunities to enhance robustness and situation-adaptable indicators through mathematical exploration. The explored solutions may be applied to multiple types of networks including disaster recovery networks.

2.1 Disaster Response Networks (DRN)

DRN comprises a list of solutions to provide post-disaster emergency responses. They are used to improve the communication capabilities of first responders to unforeseen events. In this section, we provide oversight of how DNR has been used in emergency scenarios, and the flexibility they provide to ensure the best solution is employed according to different circumstances.

In [16] and [17] the authors study the association of disaster response

network operability and sustainability with heterogeneous networks concerning the information and communication technologies they handle. They conclude by stating the importance of adequate design, not only from a technological point but also from the ease that it provides the end users to perform their tasks adequately.

In [18] the authors provide insight for communication requirements during emergency situations. They highlight how the gaps between emergency managers and other involved parties in an emergency situation is a primary deficit that must be addressed for an adequate deployment. They further provide some general foresight and recommendations on how to establish a disaster response network from a technological and logistical perspective.

In [19] the authors explore the usage of opportunistic networks to communicate important data (e.g., medical data, patience information) to a command center during an emergency response. To do so, they analyze opportunistic protocols given limited time to access the DRN network. They expand their work by studying the implications of such activities, and the applications that may be executed under such conditions.

In [20] the authors perform an overview of spectrum allocation and various technologies used for Disaster Response Networks (DRN). They further expand their research by providing information about projects regarding disaster response activities. The authors conclude their study by detailing the requirements of a DRN based on QoS, Robustness, reliability, coverage, deployment efficiency, interoperability, self-organization, and cost-effectiveness; and the influence on emergency relief operations.

In [21] the authors propose a topology based broadcast scheme to optimize user reachability, delay, and the number of re-transmissions for a DRN. They present results which suggest that their scheme adapts to the user's needs as it is communicating data. They conclude by ensuring that such results would allow a

better communication between rescuers and victims on sight.

In [22] the authors present a study where they analyze and determine the utmost important points to consider by utilizing IoT technologies regarding emergency response situations. They conclude their work by assessing how IoT enhances emergency operations by providing possible gateways and relays for a limited network. This may be possible by adapting additional modules to fixed IoT devices, where they would change their operational behavior according to the situation. However, the author does not inquire about how much technological changes or enhancements would be required to execute such a task.

In [23] the authors propose an algorithm that allows efficient energy management in cellular networks. Such application is imperative for an effective response to an emergency situation due to possible electrical outages or limited energy supply after a catastrophic event. The authors conclude their work by studying how to perform a better network recovery after an outage by considering previous outages data.

In [24] the authors analyze how hastily formed networks benefit an emergency response situation. They observe that due to their practicability and flexibility with WiFi, WiMax, and VSat technologies, it represents an adequate solution to many emergency response scenarios. They conclude their work by proposing a new organization model that includes a Human/Social domain into a DRN deployment scheme, thus providing a better understanding of how the network will operate during major activity.

In [25] the authors provide an example of a massive disaster network deployment installed in China and provide a perspective onto which points might need improvement. They expand their work by presenting an immutable need for human talent in emergency preparedness with a technological foresight to operate such heterogeneous network adequately.

Overall, DRN is comprised of different deployment practices and

technologies that allow them to be utilized in multiple emergency situations. Further into this chapter we will focus on wireless based technologies used for emergency services (i.e., Land Mobile Radios, Cell on Wheels, Satellite Communications, and Aerial Base Stations).

2.2 Land Mobile Radios (LMR)

LMR technologies are slowly being left behind as legacy technologies for voice communication in Public Safety environments. These technologies once were the utmost pillar for emergency communication systems. They provided multiple access, multiple voice communication sharing, and were robust against many external inconveniences, such as noise and signal decay [26]. To achieve this, much research was done, to an objective to accomplish a set of diverse solutions as mentioned in [11].

Within other approaches, it was possible to further extend the usage of such technology for better operations in emergency situations. In [27] the authors demonstrate how LMR networks may be used to accomplish side tasks that differ from voice communication. They study the usability of the network for location awareness during an emergency situation. They conclude their work by stating the usability of LMR technology in urban and industrial scenarios.

In [28] the authors describe a method for opportunistic spectrum access for LMR technologies. They review a successful deployment in the City of Chicago, USA, where the implementation allowed a better response to operations during a blizzard. They conclude by providing information about the increase in usability of communication channels as an indicator for their proposed method.

In [29] the authors provide a comparison between LMR and Long Term Evolution (LTE) technologies involving Public Safety communications. They review key points to consider for measuring the performance of wireless

communication technology within Public Safety Communications (PSC). They conclude their work by stating that LMR would still be needed to be the first option for Public Safety Networks (PSN), yet provide foresight of a change of technology due to the increasing demand of data consumption by first responders, which may only be provided by LTE technologies.

Currently, Public Safety Answering Points (PSAP) have now adopted or are starting to adopt data-capable LTE based technologies. This is in response to a higher need of data communication, which carries more than voice within its' data packets -usually will carry voice, location, headers (used for signaling and security), and other indicators like bio-metrics in some cases. Thus, LMR while useful for a great amount of time for public safety applications, it may not be the best solution for current need in emergency response situations for disaster relief due to limitations mentioned in [29].

2.3 Satellite Communications (SatComs) for Emergency Networks

Satellite communications have been evolving at a great pace since they parted from being used only for military purposes. Currently, aside from military use, they are also used for positioning, live event streams, data repetitions, data acquisition, photography, telegraphy, astronomy, and many other uses. This shows the capability for adaptation that such technology has for emergency response situations. In this section, we provide an overview of the satellite systems in emergency responses.

In [1] the authors provide information about different satellite-based deployments and standards regarding public safety. They include emergency message dissemination for Machine to Machine (M2M) technologies, environmental and industrial disaster alerts, authority communication safe-lines,

high data rate opportunistic aerial links, and Software Defined Radio (SDR) specifications as architectures to be used in emergency responses. In addition, they mention how the Institute of Electrical and Electronics Engineers (IEEE), the European Telecommunications Standards Institute (ETSI), and the Internet Engineering Task Force (IETF) provide standards for inter-networking opportunities in post-disaster communications. The authors conclude by providing a comparison between satellite-based communication and navigation systems currently in use, and how they assist during an emergency event. This comparison can be observed in Figure 2.1 However, for such implementations to completely work, an effective network deployment must exist beforehand for adequate operation. Moreover, some local regulation might also hinder inter-operability for heterogeneous networks.

Figure 2.1: Satellite Systems Comparison: SATCOM/SATNAV System vs. Requirements [1].

	Satellite Communications				Satellite Navigation
Feature	Full-Duplex Communication		Broadcast		GNSS
	Fixed	Mobile	Fixed	Mobile	
Aceptable Data Rates	Mid to Low Data Rates	Mid to Low Data Rates	Available	Available	Low data rates
Reception ACK	Available	Available	Non Existant	Non Existant	Non Existant
User Reach	Non Existant	Available	Non Existant	Available	Available
Mass Warning System	Optional	Optional	Available	Available	Available
CapEx	High	High	High	High	Low to Mid
OpEx	Mid to High	Mid to High	Mid to High	Mid to High	Low

In [30][31][32] the authors detail a system (WISECOM) that would enable satellite-based communications for post-disaster areas. For a solution to be viable, much work on industry needs to be considered, besides logistics for end-user devices to be distributed within an area of interest. However, they do expand their integration capabilities to other land-based systems that may be able to make use of such services. This provides an aperture for solutions that

are not geographically spread (i.e., satellite phones), but require off-site communication links.

In [33] the authors explain how different satellite architectures aid in emergency response situations. They provide insight regarding User Equipment (UE) issues and cooperative strategies for heterogeneous networks. They expand their work by mentioning positioning and message distribution services, and how they are related to satellite operations. The authors conclude by listing the advantages of deploying adequate satellite systems for future applications under emergencies.

In [34] the authors provide information about satellite-based backhaul links to extend cellular and WiFi services to users in an established location. They provide information about the need for work and research to be done regarding satellite data link balancing and logistics regarding service area focus. However, they also ensure a level of technological maturity and foresee future advancements with the adaptation of new technological advances.

In [35] the authors provide details about different approaches involving data efficient wireless links for emergencies. Wireless Sensor Networks (WSN) is mentioned as a solution to various Public Safety situations by taking advantage of size and low power consumption that the employed apparatus has. However, the limited data rates and narrow satellite-based backhauls hinder their usage in post-disaster scenarios. Mostly due to a high dependency city-grid energy and clear satellite signals due to their antenna size. This may not be a complete disadvantage, since they may still be useful by being employed by first responders on site. For example, in [36] the authors propose the usage of WSN in addition to existing heterogeneous wireless networks to extend communication services to the UE. For this, the authors portray a scenario where WSN appliances may be used as relays or for signal extension. For this to function, devices must be deployed on-site, and nearby networks need to be available.

In [37] the authors provide a study of how satellite systems can aid land-based cellular networks after a heavy disaster area where backhaul links are suddenly interrupted. They do not consider that aside from backhaul links, energy lines are also affected, and how a fully halted or intermittent energy supply may influence base station operations.

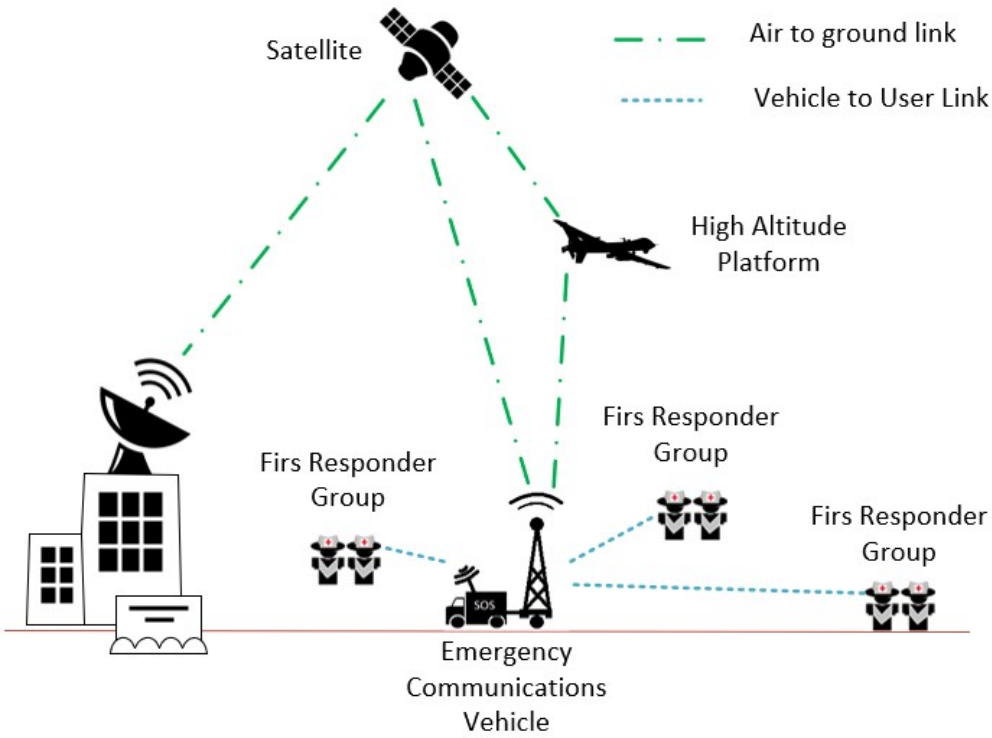
In [38] the authors propose a satellite-backed low altitude data network relay solution for post-disaster situations. They also provide information about how the technology would aid in an emergency response by optimizing the locations where ground stations must go to provide an adequate service. They do not consider the difficulty of distributing the aforementioned ground stations in a post-disaster scenario, and all the logistics it may infer.

In [39] the authors propose the use of a dual reflective antenna to assist in emergency response operations carried on site. They estimate that the proposed apparatus will be able to provide sufficient high-speed broadband for emergency situations. The equipment would still be needed to be deployed on site, and a final Radio Access Network (RAN) would be in need for a complete solution to be considered.

In [2] the authors describe a deployed project that provides high speed and reliable data to UE. The project is presented as a success story and uses Low Orbiting Satellites to provide a data backhaul to terrestrial and aerial units. The authors only consider rescuers as their final users, since the equipment necessary for the technology is not available in the market. However, the authors further expand their study by including research on the work being done to include a diverse number of devices that are employed in heterogeneous networks. The SALICE project employs different technologies which are interconnected by satellite communications. In addition, they present a High Altitude Platform (HAP) that can relay a data channel to ground stations, who will also work as relay stations for the end users (i.e., rescuers). In Figure 2.2 we can observe a

general situation of how the system works.

Figure 2.2: SALICE Baseline Scenario [2].



Satellite communications have been proven to be an adequate solution for diverse emergency responses. Although currently there are limited solutions for end users (e.g., satellite phones), there are many solutions that are satellite-based and can be extended to end users, yet distribution logistics and high cost for individual channels limit the capacity to become a final solution for a UE. However, SATCOMs are a viable solution for data relay systems in emergency responses as portrayed by [38] and [2].

2.4 Cell-on-Wheels (COW)

COWs are one of the utmost used solutions in post-disaster emergency solutions. Their ability to provide many services aside from cellphone coverage is indispensable for rescue missions. A COW, depending on the task, will usually carry the following modules:

- External communication module (e.g., Microwave or Satellite transceiver).
- Service communication module (e.g., Cellular service provider, LMR extender)
- Battery Module.
- Energy Generator (Diesel or Gasoline based).
- Security Module (Video surveillance and anti-tampering devices).
- Stabilization Mechanics (i.e., tempered lines to avoid swinging, and ground fixes).
- Control modules (These modules allow energy control and service signal control from a remote location).

During this section, we will provide some general information about how COWs are employed in emergency responses.

In [40] the authors provide an overview of parameters that emergency deployments must achieve the objective of aiding communication networks in a disaster relief situation. Parameters such as practicability, popularity, usability, capability, sustainability, operability, and adaptability are cross-referenced with different wireless technologies to asses the advantages and disadvantages of different emergency experiences. They expand by analyzing supporting functions, which include networking, scheduling, operability, and others to provide information about different approaches to existing emergency relief solutions. In addition, the authors provide information about how COWs are used in such situations.

In [41] the authors propose a mathematical model to place emergency vehicles with an emphasis on providing medical services. Such solutions are commonly related to communication networks given the need to stay connected

for telemedicine applications or emergency operational logistics such as personnel management. A COW for this solution would be attached to a medical service station for example.

In [42] the authors propose a dynamic COW allocation solution for disaster areas. Such solution oversees a possible change of population density and adapts to it. However, the author doesn't consider the logistics for operational maintenance of the COW. The authors conclude by providing information about the QoS indicator they choose to prove the viability of their module.

In [43] the authors propose to place multiple WiMax base stations to resolve a lack of coverage in a disaster zone. They do so by presenting the problem as an integer optimization problem and providing a solution to it. They conclude their work by mentioning that their solution considers a limited budget and provides a solution that will better adapt to such a scenario.

In [44] the authors provide information about a success story regarding a ground-based mobile wireless access network. The solution employs a RAN capable of providing service to all nearby UE due to its full integration built up approach that includes commercially available technologies that provide on-demand coverage and data transmission capacity. Therefore, the mentioned COW would be able to assist in rescue missions communication tasks and provide communication services to nearby civilians.

As aforementioned, a COW presents to be a viable tool in emergency responses, yet it presents some drawbacks. For example, the size and total weight the COW has made it difficult to be hastily mobilized in an emergency situation, in addition to the necessary logistics to keep the machine functioning without any problem. For example, in a scenario with no external energy resources, a COW would require constant fuel refilling for the generator to keep functioning, or a rapid alternative energy installation (e.g., solar panels). Both solutions require much work and may only be deployed given specific

environmental, and security conditions. Therefore, a COW might not be a viable solution for a primary stage emergency scenario.

2.5 Device-to-Device (D2D) Communication Solutions

Device-to-Device Communications became an official part of LTE starting from release 12. This technology allows a UE to directly communicate with another UE with partial or no participation from a Base Station (BS). In release 12, D2D was enabled solely for public-safety-enabled UE, which are UE that may be only operated by first responders. They have the capacity of performing nearby UE discovery procedures, a direct device to device communications, group communications within an UE cluster, and message dissemination between inter-connected groups. For release 15 every UE will be able to perform device-to-device communications. However, it will be a slow adaptation given the push-back that industry is presenting to this standard. In this section, we will provide an overview of device-to-device technologies and how they can provide in emergency situations.

In [45] the authors analyze key indicators that would allow D2D communications to be available for emergency situations. The authors extend their work by analyzing a scenario where there is limited to no coverage for a certain amount of UE and analyze how they can communicate by partial use of a nearby BS. They propose a concept of a network that would allow UE to dynamically form clusters by implementing cluster-heads which would be in charge of administrating the group. They conclude their work by analyzing D2D synchronization and communication protocols, and how they would be adapted to the proposed idea.

In [46] the authors provide information about the evolution and highlighted parameters of D2D communications. They analyze how interference

management, resource management and optimization, performance evaluations, and the applications and services provide a function within D2D communications. They further extend their work by analyzing D2D test-beds used for standardization, and how the current standard is assembled.

In [47] the authors proposed an out-of-coverage model and a relay selection method based on the RBC (Remaining Battery capacity) of the relay candidate. It concludes by presenting results based on a simulation considering a fixed energy consumption from each device circuits in addition to the necessary energy consumed by transmissions based on the distance between UE.

In [48] [49] the authors proposed a model to select a cluster head that is currently out-of-coverage range bordering a Base Station (BS) range perimeter. This head cluster is portrayed in different scenarios by which different solutions are proposed. Finally, the authors select a cluster head based on an optimized solution that considers distance, other nodes, and remaining battery life.

In [50] the authors provide information about the potential advantages of D2D in disaster situations. They analyze the optimization of spatial density and transmit power capacity for a multi-hop D2D system in a non-functional area with the help of a relay assisted network, and provide information about the shown performance according to power efficiency indicators. They conclude by stating that the architecture enhances D2D performance in networks from a system and power efficiency perspective.

In [51] the authors proposed a Hybrid D2D architecture that functions within local and cloud controllers that would enable scalability and efficiency when trying to enable an emergency network. The authors mention how such a solution would aid rescuers with victim localization tasks through a Software Defined Network (SDN). They conclude their work by stating that their tests comply with the storage and power limits that WSN presents, given that the solution is thought to be applied in a WSN enabled environment.

In [52] the authors' propose a system that would use multiple UE to relay a message to a destination without the intervention of a central BS. They analyze the delay, throughput, and many connections indicators provided by their system and conclude that it would be an adequate addition for emergency usage.

Moreover, for D2D communications to provide adequate service in such scenarios, UEs always have to be connected and communicating between each other, and a working BS must be nearby to supply communication services to a cluster gateway or relay node. Such approach would become a problem in a scenario where the UEs are spread-out, since a cluster may be completely isolated, or depend on a single device for communications at one point, which creates a single failure point scenario and bottleneck problem. In addition, the available radio resources and the remaining battery lifetime for a User Equipment (UE) has a heavy influence regarding it's functional time, which may cause cluster segmentation, and thus, create a need for iterative re-discovery procedures and re-configurations within a formed mesh network. Therefore, D2D communications may be useful in a cell border-line out of range event and as a range extension solution; yet it would not be an adequate solution for a scenario with UEs spread along large areas.

2.6 Unmanned Aerial Vehicles (UAV) Solutions

A different approach for a large area with spread UEs within the primary stage would be implementing Unmanned Aerial vehicles (UAVs), where each of these devices may be able to provide a signal for communication purposes. Such technology is referred to as Unmanned Aerial Base Stations (UABS), where the UAV's objective is to relay communications or provide intercommunication services between two or more parties. Such techniques are still under research given the many challenges they portray (e.g., flying time, battery life, coverage).

In this section, we describe some UAV solutions and how they may influence emergency scenarios in need of communication.

In [53] the authors study many types of UABS and list their advantages and disadvantages depending on the type of aerial vehicles. They consider drones (quadcopters), aircraft (gliders and motorized), airships (that carry a passenger or payload), and their model (a glider air balloon) as subjects of study. In their study, they consider various parameters, such as payload weight, optimum altitude, technology compatibility, expertise required, and communications related parameters. They conclude by determining the scenarios by which some aircraft may have a better performance, and how their proposal outperforms some of them given a specific scenario.

In [54] the authors analyze how UABS may enable different types of missions. For this, hey consider MIMO, ground stations, device relaying, and D2D communication with a focus on UAVs. They study how some tasks can be accomplished by executing them in different stages. For example, when a UAV circumvents a sector to distribute a message, and afterward all ground devices communicate via D2D. They conclude by noting the advantages that such technologies may provide, such as data relays, information dissemination, and coverage assistance.

In [55] the authors propose a density-aware UABS deployment suitable for disaster discovery scenarios. They propose a clustering algorithm for terrestrial stations that would access a Low Altitude Platform (LAP). They conclude by analyzing how their algorithm outperforms a Hybrid Energy-Efficient Distributed (HEED) algorithm by utilizing energy performance as a performance indicator.

In [56] the authors explored the usage of a hovering UAV along with low and high power BS to satisfy QoS requirements in a 5G network using mmWaves for Public Safety Networks (PSN). This topic has been studied from many perspectives to provide the best solution for a drone placement. The authors

expand their work by discussing routing protocols, Cognitive Radio (CR), and deployment operations where D2D technologies and UABS are utilized simultaneously. The authors conclude their work by stating how the UABS in addition to usual cellular infrastructure improve service without sacrificing Quality of Service (QoS).

In [57] the authors propose a method where UAVs may be used as data relay nodes as UABS and study the communication challenges involved in deploying such applications. They focus their research into coverage, relaying, and message dissemination. They conclude by highlighting a key performance indicator based on mobility.

In [58] the authors proposed a method to deploy several UAVs within a given area. For their work the considered inter-drone interference, area coverage,beam-width of a directional antenna, and altitude. They conclude their work by proposing a deployment based on the circle parking theory and provide data regarding how the method provides maximum coverage while utilizing a minimum of transmitting power for every drone.

In [59] the authors study how achievable power and sum-rate gains provide an optimum height for an UABS. They conclude by stating how favorable altitudes would provide a more efficient deployment in comparison with other analysis in the field.

In [60] the authors suggested the usage of a bisection search algorithm to determine the best position for an Aerial BS based on suburban, urban, dense urban, and high-rise urban environments. The proposed system maximizes the number of users covered by a UABS. The authors conclude by stating that their system may be useful in scenarios where there is congestion or lack of service.

In [61] the author presents a solution for UABS placements based on energy efficient considerations. The considerations taken for this are area coverage and transmission range. They conclude by enabling plane differentiated simulations

and yielding results that show savings in transmission power and an increase of the number of covered users for heterogeneous cases.

In [62] the authors provide an optimization solution for a Low Altitude Platform (LAP) given some hovering drones and destiny service areas that need coverage. They propose WI-FI as their transmission technology and focus their results on such a scenario. In addition, they state that the altitude of a UABS will not considerably affect their coverage area. The authors conclude by recommending altitudes given their assumed scenarios.

To expand on previously explored solutions, research has been done for drones that hover in different spots given different conditions. In [63] the authors analyze a multi-rotor drone distribution problem and provide different algorithms to solve a Travelers Salesman Problem (TSP) based on energy cost, schedule, arrival times, and drone cost. The authors conclude by stating the importance of optimizing battery weight and drone re-utilization for a successful deployment.

In [64] the authors proposed a TSP-based method for serving nodes (i.e., package delivery) with a moving truck and a moving drone by synchronizing the drone's departure from the truck, serving time, and scheduling the return at a rendezvous point to optimize total travel time. Such solutions will usually be based on the TSP or a Multiple TSP (MTSP), to which researchers have also made progress.

Further research has been done considering drone movement and additional constraints to provide a different approach to new problems and scenarios. In [65] the authors proposed a control model for a drone so it may move at a fixed altitude to improve spectral efficiency in a given system. In addition, they provide some deployment and simulation results by implementing practical constraints to their optimization problem.

In [66] the authors proposed a method to dynamically re-position hovering drones to improve spectral efficiency. They study their method by adjusting

speed and considering energy consumption and achieve an optimized position for the drone accordingly.

In [67] the authors proposed a method to calculate dynamic routes between different clustered unserviced UE. These routes are constantly updated, and the drone will commute and hover from point to point. They consider a single channel, height, transmission power, and interference to calculate their results.

In [68] [69] the authors proposed a moving drone solution for global message dissemination, a network bisection, and a k-connectivity problem by solving a Spanning Tree solution between established points and providing such solution as a path.

In [70] the authors analyzed a long-range route planning solution based on communication requirements and use a spanning tree solution to provide the best route.

In [71] the authors propose a spiral placement algorithm to provide signal coverage to a sector. They compare their work against other schemes that yield a number of Macro Base Stations (MBS) to provide full coverage of a sector. They conclude their work by stating the benefits of their algorithms and listing future marks to be reached for backhaul technologies.

In practice, there are many successful cases of real-time usage of these types of equipment. Although much work still needs to be done in the area to achieve all goals and reduce many of the disadvantages that UABS present. In [72] the authors provide a Moving Aerial Base Station capable of providing services to any UE by utilizing a backhaul link. The UABS provides a RAN by employing 3G, 4G, SatComs, and Wifi technologies with different reachable radius.

In [73] the authors analyze the overall usage of UAVs for emergency information and communication management during an emergency scenario. They list various challenges and advantages that occur when implementing UABS for DRN. They further expand their work but stating the requirements

needed for deployment given different natural hazards, and listing the usage of an UABS depending on the disaster stage (i.e., preparedness, assessment, response, and recovery).

Other solutions may be implemented by swarming the airspace with drones if conditions allow. For example, [74] [75] [76] study how a group of UE or ground devices may be serviced by multiple drones, given a MIMO access technology is implemented. They also provide multiple performance measurements to ensure such deployments work as expected.

Within the studied solutions, the main focus was to achieve a better result from UAV limitations. Provided the pace of technological advancement, it is also acceptable to consider some of the constraints as solved for future deployments.

2.6.1 Spectrum Efficient (SE) UAVs

In this section, we survey some work done by analyzing SE, and its' influence on the service provided by UABS.

For the following study, the authors propose a system model where a Poisson distribution sets the number of users, and their cellular service is provided by a Macro Base Station (MBS). Afterward, they simulate a halt of coverage in some areas as a product of a natural catastrophe. To this, in [77] they provide a solution to deploy multiple UABS to provide coverage based on a performance indicator of the 5th coverage percentile of spectrum efficiency given a fixed threshold. The solution to the optimization problem is then obtained by implementing a brute force algorithm. In [78], they enhance their solution by implementing a Genetic Algorithm (GA) based an iterative number of UABS, and providing a result as the best solution to the given scenario, where a certain amount of hovering UABS will be placed to provide coverage for an area. In [79] the authors proposed an optimization algorithm to enhance aggregate capacity based on their Spectral Efficiency requirements for Heterogeneous Networks

(HetNet) derived from Further Enhanced ICIC (FeICIC) and REB. Finally, in [80] they further enhance their study by extending the length of their chromosomes for their GA to include a scheduling threshold for MBs and UABS, a power reduction factor, and a Cell Range Expansion (CRE) factor while considering ICIC and FeICIC. As a result, their solution provides a great amount (above 95%) of sector coverage. The solutions provided are for static hovering drones on a static altitude and fixed base transmission power. However, the solution does not consider what happens with unattended sectors, or a physical topology change as a result of some UE performing minor geographical position changes while seeking to obtain better service. This reaction will reduce the SE offered by the Macro Base Station of Interest(MOI) or Unmanned Aerial Base Station of Interest(UOI) to a set of UE given the increase of density in an area, thus creating breathing cell effect [81].

In [82] the authors study the influence dynamic re-positioning of a UABS due to geographical positioning uncertainty of UE. By doing so, they may serve more users and reduce their signal blocking probability, thus increasing SE. The authors propose two algorithms that would enable a UABS at the height of 10m to reposition itself depending on the user's mobility and activity. They conclude that by repositioning the drone, they were able to double SE while maintaining the same level of energy consumption.

The solutions provided insight into how UABS are used in different scenarios. While they may present many drawbacks, the opportunities presented in foresight are much greater with various solutions that may be enabled given technological advancements. Furthermore, considering their capability to fly over uncertain terrain, provides a major advantage over other solutions given a primary stage response scenario.

2.7 Traveler Salesman Problem (TSP) and Genetic Algorithm (GA)

Within the explored solutions, some are presented as optimization problems that may require meta-heuristic solutions. In this subsection, we provide a brief insight of how solution models based on GA for a TSP.

In [83] the authors proposed a solution to the Multiple Traveler Salesman Problem (MTSP) through a GA (Genetic Algorithm) method by using a single chromosome and simple crossover and mutation. In [84] the authors presented a solution to MTSP with GA by using multiple chromosomes and employing in-route mutations and cross-mutation. In [85] [86] the authors proposed a solution to the MTSP by using a node chromosome and a salesman control chromosome. The first authors further enhance their solution by employing a modified Simulated Annealing Genetic Algorithm. In [87] the authors propose a solution to the MSTP problem by employing a gravitational system.

From the review, GA is proven to be a valuable tool to find an optimum solution by avoiding local solutions and the use of brute force algorithms. A tool that would be favorable for use given the challenges presented when analyzing placement problems with UABS.

2.8 Summary

Previous studied research work and deployment applications have provided insight regarding technologies used for a DRN. In continuation, a summary list of advantages and disadvantages analyzed during the review will be listed per surveyed technology.

LMR

Advantages

- Reliable for voice communications.
- Long range communications.
- Mature.
- Resilient to external interference.

Disadvantages

- Very limited data communication.
- Only proprietary solutions are available.
- Expensive to deploy and maintain.
- Incompatibility with other technologies.
- Slowly being left as a legacy technology.

SatComs

Advantages

- Worlwide Coverage.
- Many available communication solutions for emergency situations.
- Mature and successful deployment examples.

Disadvantages

- Requires special devices to access the service.
- Limited Bandwidth available.

- High expenditure and operational costs.
- May be difficult to access in forested or urban areas.

COW

Advantages

- Can be mobile given adequate conditions.
- Can be set in a reduced amount of time compared to a real cellular tower.
- Can be equipped with multiple technologies.
- Provides customizable heights for antennas.
- Can be changed places.
- re-utilizable for different missions in different places.
- Endures though external factors (e.g., heavy weather).
- Can provide high data rates an coverage to a zone.

Disadvantages

- Heavy.
- Difficult to assemble.
- Slow transportation.
- Requires constant maintenance.
- Require external energy sources (i.e., energy from the grid or combustible for a generator).
- Requires communication backhaul.
- Requires stable ground for installation.

D2D

Advantages

- Can work with partial or no BS intervention.
- May be used to form Ad-Hoc networks when needed.
- Provides extended coverage.
- Provides discovery procedures, one-to-one, and one-to-many communication.
- Standard currently enables any UE to use the technology.
- May be used in multiple solutions (e.g., range extender, relay, store-and-carry solution).

Disadvantages

- Requires a gateway channel for external communication.
- Requires a gateway for external communications.
- May have a single point of failure (e.g., a cluster head with no successor).
- Depends on UE battery, antenna, and other hardware limitations.
- Currently may only be used by first responders.

UABS

Advantages

- May be deployed from a location remote to the interest zone.
- Come in many solutions.

- Can increase service area by adapting to an optimum altitude.
- May be deployed in adverse environments (e.g., in geographically though terrain).
- Many solutions are available for various type of DRN scenarios.
- May offer additional services aside from communications (e.g., video vigilance, Light Radar (Lidar) terrain sampling).

Disadvantages

- Prone to bad weather conditions.
- deployment limited by the amount of fuel or energy that the UAV can carry.
- Complex deployment logistics.
- Requires advanced training for adequate deployment.
- Equipment might require constant refueling or a frequent battery change.
- Susceptible to great amounts of damage after a fall.
- The higher the used technology, the more expensive the equipment.

Most previously described research work focuses on securing a fixed point to provide a wireless communication service for UE. LMR, while currently heavily deployed, usually will have few BS due to the frequency they use and the limited service they provide. These aspects would limit the network to single point failures and to provide a mostly voice-only service. In addition, it requires specialized equipment to be used; something that would not be available to civilians.

Satellite communications require highly priced and specialized devices to access the service. While there may be viable solutions for the given scenario, a deployment would also require heavy logistics to ensure that spread civilians have access to the network.

A COW presents itself to be logistically difficult to implement during the first days of an emergency. This is because many aspects have to be met (e.g., right geographical settlement, available fuel or external energy, clear paths, etc.) for adequate installation.

D2D is currently limited to public safety UE. While it is foreseen that every UE could soon be able to access a D2D network soon, it presents some challenges considering single points of failure for a gateway, and the need of multiple UE to be always near each-other for practical communication.

Hovering UABS still present some challenges on logistics and the amount of energy needed for adequate long-lasting deployment. While single path routes Aerial Base Stations have only been studied considering a one-way fly path only (As data recollection systems). Such examples would not suffice the required needs for periodic communications needed for UE in a post-disaster emergency situation.

As far as this research is concerned no earlier studies have been conducted regarding a cyclic path coverage that would provide periodic or intermittent communications given a primary phase deployment situation. This thesis will focus on determining a deployment solution for MABS. The multiple path generation solution will consider a periodic service capacity (i.e., 200 Kbps) supply to geographically spread users with no service. Such paths would provide a more realistic scenario for aerial moving vehicles with limited or temporary propulsion (e.g., hot air balloons, gliders) to increase air time, thus achieving resource efficiency and extending periodic service time before the need for replacements or refueling. This will be achieved by implementing a combination

of genetic algorithm, and graph theory analysis. Within the genetic algorithm, fitness functions that consider serviced UE, angles, traveled distances, MABS path intersections will be employed to ensure the best solution.

Chapter 3

System Model and Mobile Aerial Base Station Deployment Schemes

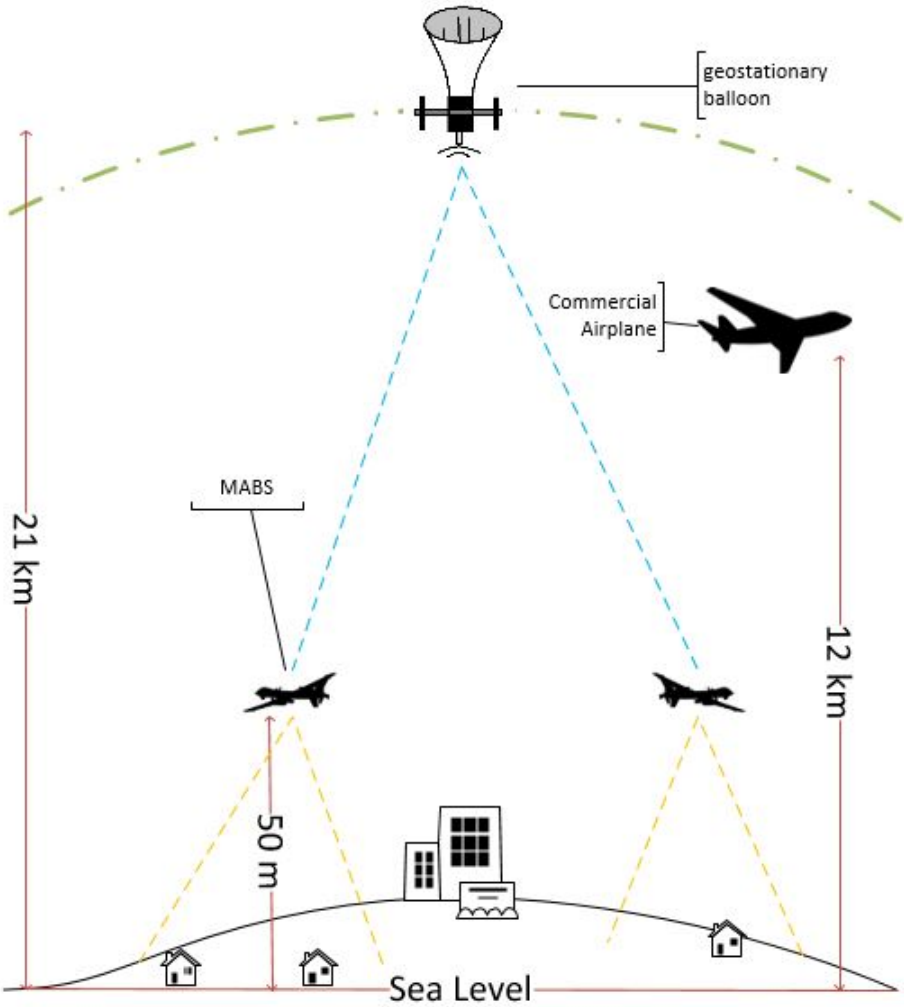
In this chapter, we describe the assumptions, the system model, and the proposed solution. We consider a problem for a primary-stage post-disaster emergency scenario. We further explain the detailed solution for Mobile Aerial Base Station (MABS) deployment based on the algorithm using Affinity propagation clustering, graph theory, and genetic algorithm.

3.1 System Model and Assumptions

One of the objectives is to provide momentary connectivity for UE. To accomplish this, a general system comprised of Mobile Aerial Base Stations communicated thorough a back-haul supported by connected geostationary communication balloons (GCB), satellites, and commercial airplanes will be assumed. A depiction of the mentioned system can be seen in Figure 3.1, where the dash-dotted line describes the geostationary altitude where a GCB will be positioned and constantly communicating with a ground station. With a similar idea, a satellite or a commercial airplane may be able to provide a back-haul service to the MABS, which is signaled with a dotted-dashed line. The arrows on the sides describe the flying altitude of each element. The dotted coming from the MABS line refers to the coverage provided by a MABS. Further, into this

work, the focus will be onto the UE on land (population represented by density in accordance with the placed building) and the MABS.

Figure 3.1: Back-haul Communication System for MABS



A set of UEs will be represented by a distribution of points in a $25Km \times 25Km$ area, which resembles a small city. The number of UEs will be calculated by using:

- **Normal (Gaussian) Distribution:** Where the mean will be the desired density x the area.
- **Uniform Distribution:** With a range of 1 to 100 for UEs, and 1 to 10 for BS. And a $\sigma = 10$.

UE	User Equipment
BS	Base Station
λ	Density value for Poisson Distribution
bps	Bits Per Second
C	Capacity in bps
Bw	Bandwidth for a channel in Hz
SNR	Signal Noise Ratio
PL	Path Loss
A	Frequency Incidence factor
B	Base Station Antenna Height factor
$a(h_m)$ and Env	Environment Correction factors
f	Frequency in Mhz
h_b	Height of the Base Station
h_m	Height of the Mobile device

Table 3.1: System Model Notation Table

- **Poisson Distribution:** With $\lambda = (100 \times 25)$ for UE, and $\lambda = (10 \times 25)$ for BS.

This will allow representation of different possible UE distribution on PSC situations.

The capacity requirement for each UE will be equal or higher than 200 Kbps, and calculated using the Shannon-Capacity formula (3.1) with a bandwidth of 1.4 MHz; where C is the capacity in bps and Bw is the bandwidth in Hz.

The SNR is calculated with (3.2), where P_i is the power that the MABS or BS receives from an assigned UE, and P_j is the power that a BS or MABS receives by not assigned UE, plus a ground noise η set to -112 dB [79]. For these calculations, the BS power is set to 46dB, a MABS is set to 30 dB, and a max of 10 dB for a UE [77] [78] [79] [80]. MIMO access methods are assumed for multiple UE assigned for a single UE.

$$C = Bw \times log_2(1 + SNR) \tag{3.1}$$

$$SNR = \frac{P_i}{\sum_j P_j + \eta,\ \forall j \neq i} \tag{3.2}$$

The propagation model to be used will be an Okamura-Hata common curve fitting model to provide the utmost approximate representation of a general scenario [81] [88]. The carrier frequency will be of 700 MHz (LTE Band 14).

$$PL = A + B \times log(d) + Env \tag{3.3}$$

$$A = 69.55 + 26.16 \times log(f) - 13.82 \times log(h_b) - a(h_m) \tag{3.4}$$

$$B = 44.9 - 6.55 \times log(h_b) \tag{3.5}$$

And considering our small city scenario.

$$a(h_m) = (1.1 \times log(f) - 0.7) \times h_m - (1.56 \times log(f) - 0.8) \tag{3.6}$$

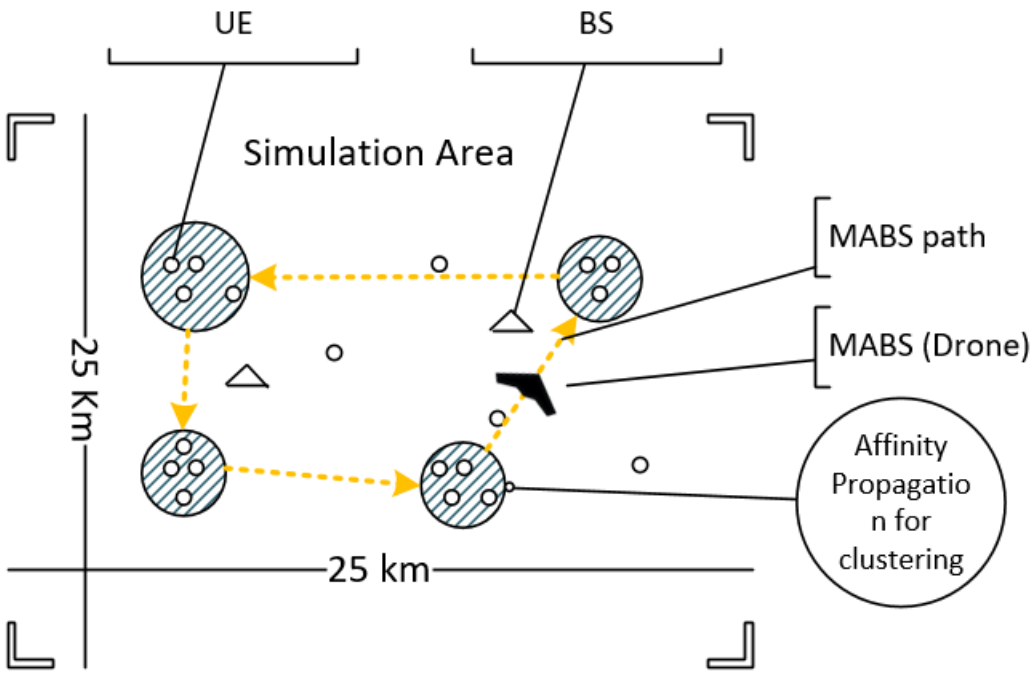
$$Env = 0 \tag{3.7}$$

Each MABS will be considered to be flying at a constant speed of 50 Km/h, and no further external interference (e.g., wind speed, weather condition, natural fauna) will be considered for the study.

A set of UE with no service will be clustered together with an Affinity Propagation algorithm [89] [90], and implementing a 0.6 as the damping value.

The described model can be found in Figure 3.2. Here UEs are represented by small circles, and the cluster they form through Affinity Propagation is represented by a larger circled with diagonal line filling. The dashed arrows are the path a mobile MABS must follow to provide the desired service. The whole

Figure 3.2: Proposed System Model Visualization



experiment will be executed within a 25 KM square area. Furthermore, the mobile MABS will be providing temporal communications to all UEs within the desired cluster; in addition to an UE it might encounter in its path while traveling from a centroid to another.

3.2 Problem Formulation

The model will be represented as an MTSP with no common origin point. The considered graph set will be $G = (V, E)$, where V represents the set of cluster determined by the UE density in certain areas, and E edges will be the distance between each cluster center (V). Then the model may be represented by the optimization function on (3.8). Where it is sought to minimize the number paths. In (3.8),(3.9),(3.10),(3.11) x_{ij} represents whether a drone will visit a cluster $\in V$, c_{ij} represents the cumulative capacity requirement; where cumulative capacity (C) is the sum of all capacities from UE in that cluster. (3.9),(3.10) represent a constrain so that each drone may visit one node only.

(3.11) limits one path per cluster point $\in V$.

Optimization Function

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \times x_{ij} \tag{3.8}$$

Subject to:

$$\sum_{j \in V(k,j)}^{m} = s_k \tag{3.9}$$

$$\sum_{j \in V(j,k)}^{m} = s_k \tag{3.10}$$

$$\sum_{j=1}^{m} x_{ij} = 1, \forall_i \in V \tag{3.11}$$

The MTSP problem then transforms to (3.12) under the consideration that a path must at least have one MABS (Mobile Aerial Base Station) in it. A MABS for this work will be considered a remotely piloted or autonomous non-hovering flying vehicle which does not require a human passenger to be flown (e.g., glider, hot air tactical balloon).

Optimization Function

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{u=1}^{N_{MABS}} c_{ij} \times x_{iju} \tag{3.12}$$

Where N_{MABS} is the total number of MABS available for the mission, and x_{iju} is a cluster center cardinal point which has been assigned as a path point for an MABS (u). Furthermore, each iteration of c_{ij} represents a cluster center which has a particular requirement for a certain amount of capacity, and x_{iju} can be visited by only a single MABS. Because we consider that each UE only requires a moment of connection to perform a quick messaging activity,

connection time may be small, but it must be repetitive. (3.9),(3.10),(3.11) are included as constraints.

Since a cycle is considered, no original point of departure may be placed. This can be interpreted so that one path may have one point of departure or all points within a path can be considered to be as starting points. Therefore the total number of edges E is represented by (3.13) [91].

$$total(E) = \frac{n!}{(n-2)! \times 2!} = \begin{pmatrix} n \\ 2 \end{pmatrix} = \frac{n(n-1)}{2} \tag{3.13}$$

For each path, there will be an isomorphic graph, since $A \rightarrow B = B \rightarrow A$, but a fly-ability parameter must be considered, a turn is represented by a connection of points $A \rightarrow B \rightarrow C$, and the angle ABC may differ depending on the path solution. For example, in a set A, B, C, D distributed in a perfect square, it will not be the same to commute $A \rightarrow B \rightarrow C \rightarrow D$ to $A \rightarrow D \rightarrow C \rightarrow B$ because of a larger distance and narrower angles [91].

Moreover, a best effort on path selection is expected regarding that there may be more MABS available, this is a response to a need to either reduce the amount of disconnection time for a UE, or more devices for replacement in case of need.

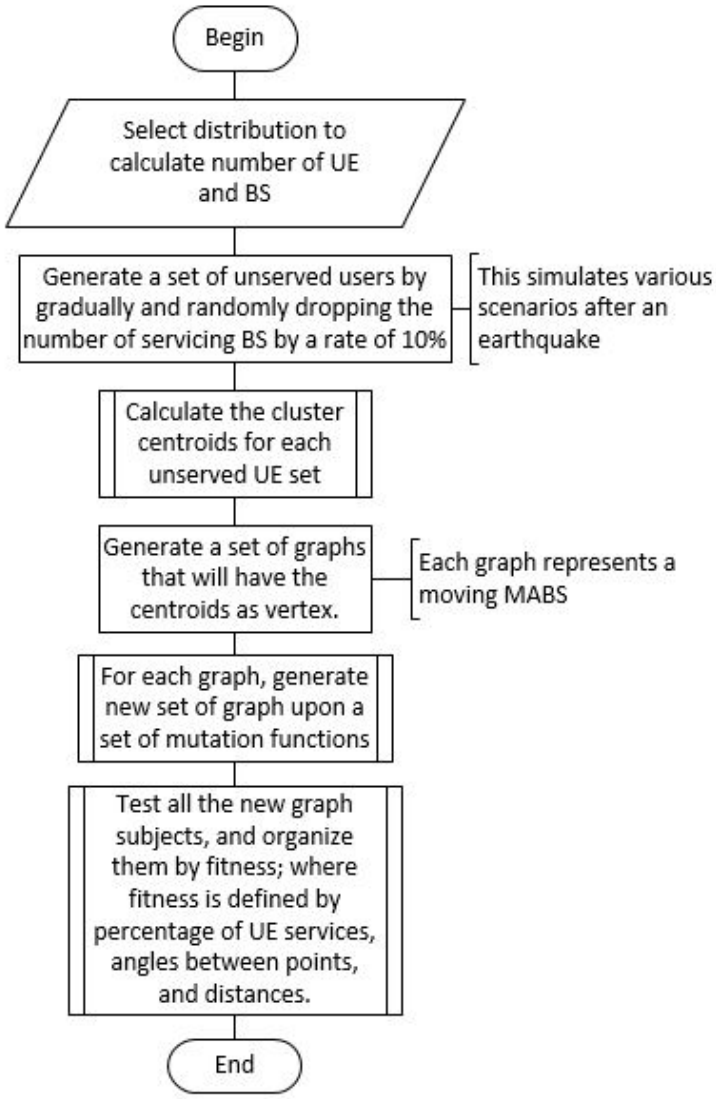
3.3 Proposed Algorithm

To provide a solution to the problem, a Metaheuristic algorithm will be implemented. A Genetic Algorithm (GA) is a process based on how evolution and natural selection works in nature. Here, different Evolutionary Objects (from now on considered as specimens) with one or multiple chromosomes by process of mutation and reproduction allow the yield of an improved specimen better adapted to current circumstances [92] [93]. The chromosomes for each

specimen are a combination of parameters that can be changed to provide the best solution to a proposed problem. At a point, all specimens must be tested to acknowledge the specimens. This process is called fitting, where the functions and equations by which each specimen is measured or tested are based upon constraints or requirements for a determined system.

In Figure 3.3 a diagram flow of the proposed algorithm is shown. Further explanation of the work can be found in the rest of this chapter.

Figure 3.3: Proposed Algorithm Diagram Flow



3.3.1 Affinity Propagation Clustering

Categorized as a non-supervised machine learning method or clustering method, affinity propagation presented to be the utmost useful method to perform UE-clustering provided the geographically spread UE in need of a clustering center based on density [94]. Initially, k-means was considered due to its simplicity and effectiveness but was later discarded as the method requires a number of clusters in which to classify the unserviced UE. Affinity Propagation is based on finding a balance between a responsibility matrix and availability matrix, which is formed by a broadcast message from each node to its' neighbors. The responsibility matrix represents how a k node should represent a node i. The availability matrix has information about how much node i wants to be represented by node k [95]. The process will usually evolve from a scenario where all UE are candidates for representation, mid-way into sectorization due to proximity, to a final stage of no change, thus providing cluster centers [89].

3.3.2 Specimen Chromosomes

In GA, chromosomes are the part of a specimen that must be mutated, or combined with other chromosomes to produce more and possibly improved off-springs [92] [93]. Once a set of UE has been categorized into clusters, a graph represented by (3.14) (3.15) are obtained; where V are the centroids for each cluster, and E represent the Euclidean distances between clusters where i and j represent cardinal points.

$$G = (V, E) \tag{3.14}$$

$$E = \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}\right) \ \forall\, i, j \ \in V, i \neq j \tag{3.15}$$

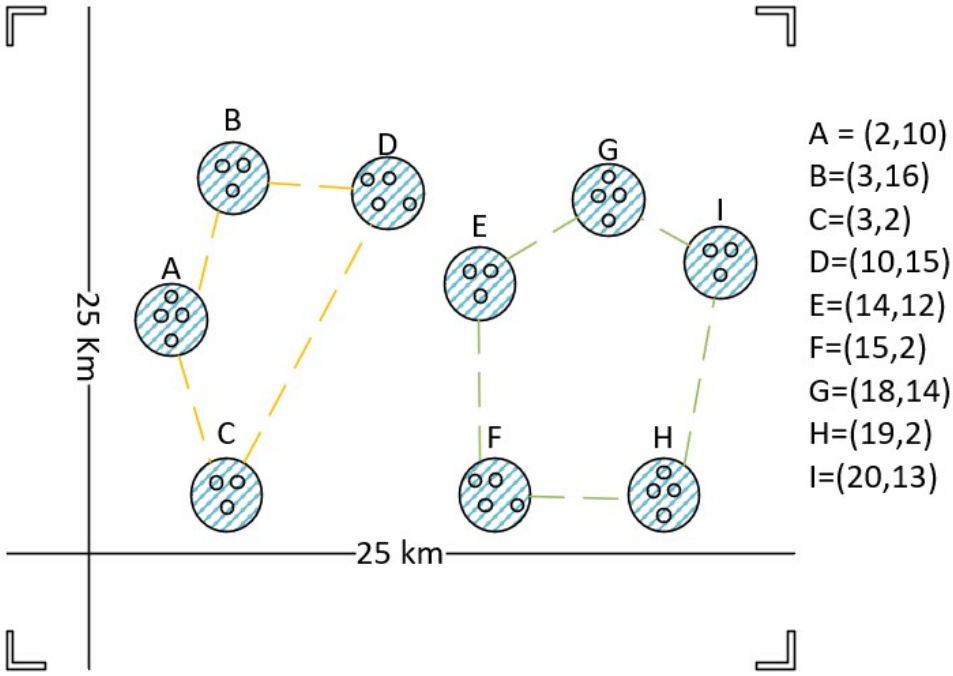
A chromosome will now be represented by multiple slots, where each slot

represents a cluster point as stipulated in (3.16); where the length of a chromosome (m) must be at least 3.

$$chromosome = [i_2, i_2, i_3, ..., i_m],\ i \in V \tag{3.16}$$

In Figures 3.4 and 3.5 we can appreciate an example of an specimen with two chromosomes. Each chromosome is an independent path with different points as previously described. Chromosome 1 is a path that passes over clusters A, B, D, C, and chromosome 2 is a path passing over clusters E, G, I, H, F.

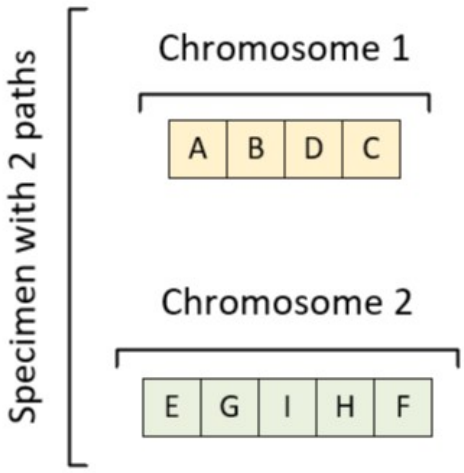
Figure 3.4: Chromosome Path visualization



3.3.3 Mutators

Mutators allow a set of specimens to proliferate within given limitations and in an orderly manner. Each specimen will then produce between three and six new specimens, depending on the order and number of chromosomes. By implementing such mutators, it's expected to obtain the best specimen by a posterior fitness selection, and only proliferating the best-fitted specimens [92]

Figure 3.5: Chromosome Construction visualization



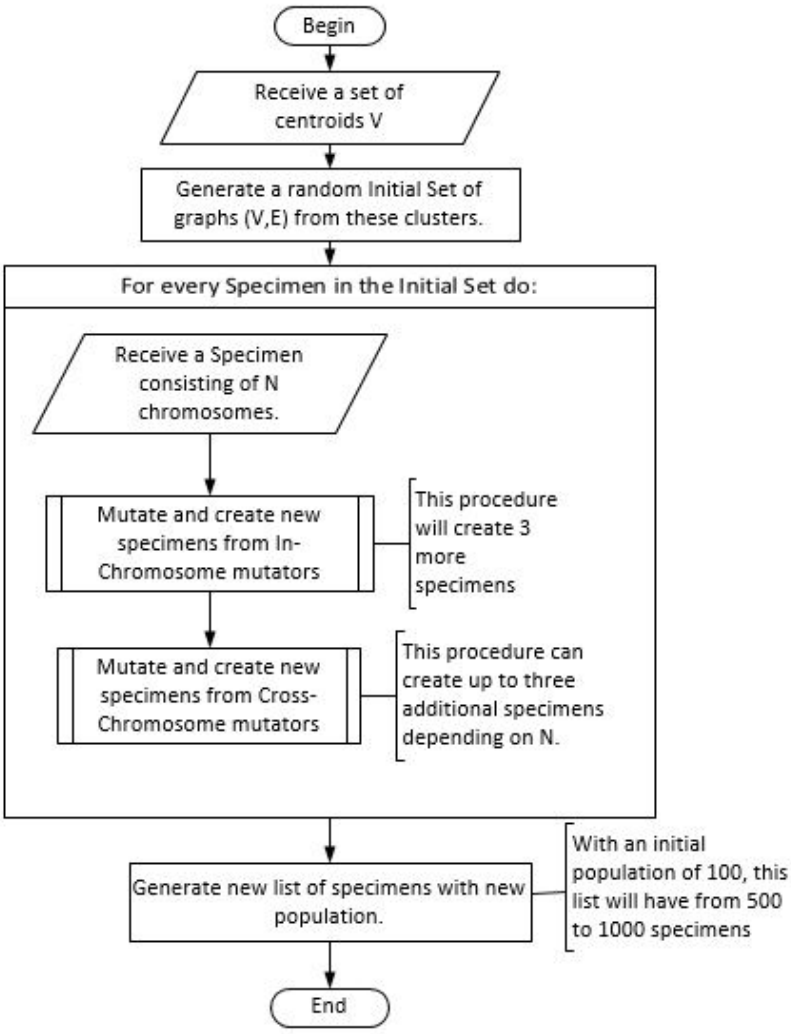
[93]. Cross-Over mutators will not be applied since the number of chromosomes per specimen will vary, thus not allowing an effective chromosome cross-over method [96]. Furthermore, for any random selection procedure within the mutation functions, a uniform random distribution will be utilized.

In Figure 3.6 a diagram flow of how the mutation functions are applied can be found. Here, it further explains how from a selected specimen, a range of four to six new specimens can be created. The mentioned Initial Set of graphs (V, E) are created by determining how many 3 point chromosomes can be created from the total number of cluster centers. If the total number of cluster centers is not a divisor of three, then the remaining cluster centers will be distributed between the first and second created chromosome. Thus, resulting in a set of chromosomes (of size three or four) which form a single specimen.

3.3.3.1 In-Chromosome Mutators

This set of mutators will produce a new specimen while maintaining the length of each chromosome the same [92][93]. These mutators will then achieve a goal of seeking the best combination of cluster center visiting order while maintaining the same amount of required drones. For these mutators, a new specimen will be

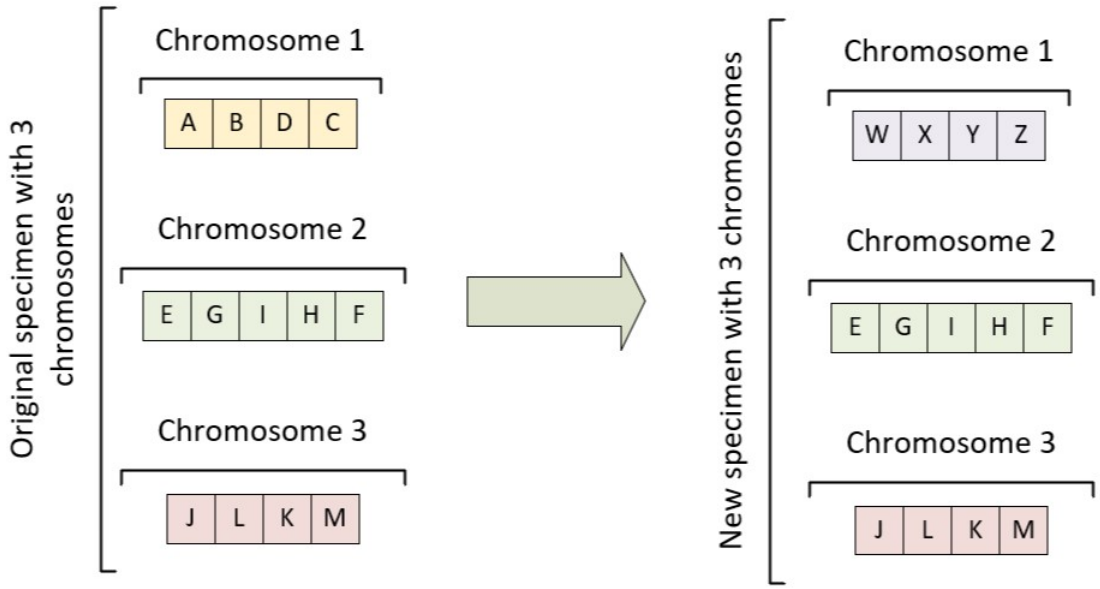
Figure 3.6: Mutators Application Diagram Flow



produced with all the exact number of chromosomes, minus the selected chromosome, and the new mutated chromosome as it can be observed in Figure 3.7, where only the first chromosome was changed, thus the specimen maintains the same number of chromosomes.

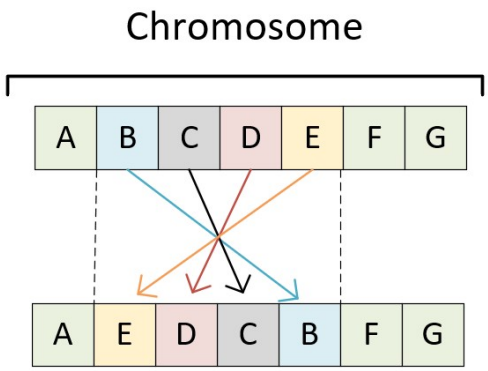
In-chromosome section swapping mutator From the selected specimen, a random chromosome will be selected. Afterward, a section of that chromosome will be selected for swiping, thus generating a new chromosome and a new specimen [96]. In Figure 3.8 it can be appreciated how the points B, C, D, E are selected as the section to be swapped, and the result of the new produced

Figure 3.7: Specimen Creation from an In-Chromosome Mutator



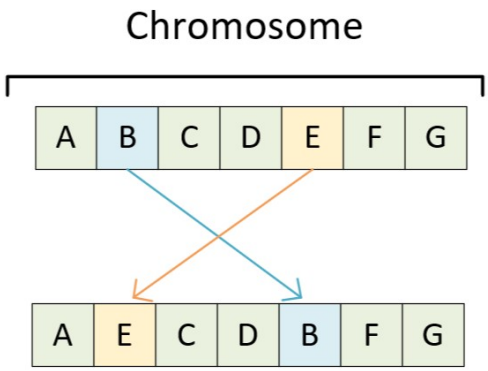
chromosome which yields a new order combination.

Figure 3.8: In-Chromosome Section Swapping Mutator



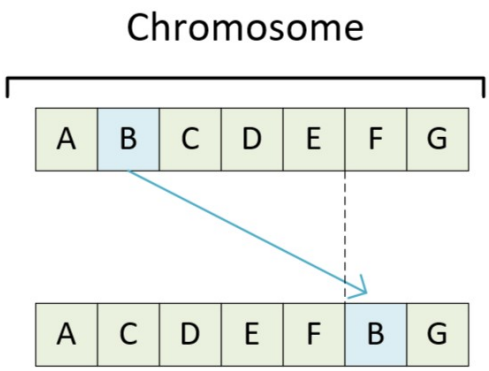
In-chromosome point swapping mutator From the selected chromosome, two points will be randomly chosen and swapped [96]. In Figure 3.9 it can be appreciated how points B, E are selected for swapping — thus resulting in a new specimen with a minor combination order than the one obtained from the swapping mutator.

Figure 3.9: In-Chromosome Point Swapping Mutator



In-chromosome point insertion mutator From the selected chromosome, a single point will be removed and further inserted in another position [96]. In Figure 3.10 it can be appreciated how point B is moved four spaces ahead. This creates a new chromosome with not much affectation from its original chromosome.

Figure 3.10: Chromosome Point Insertion Mutator

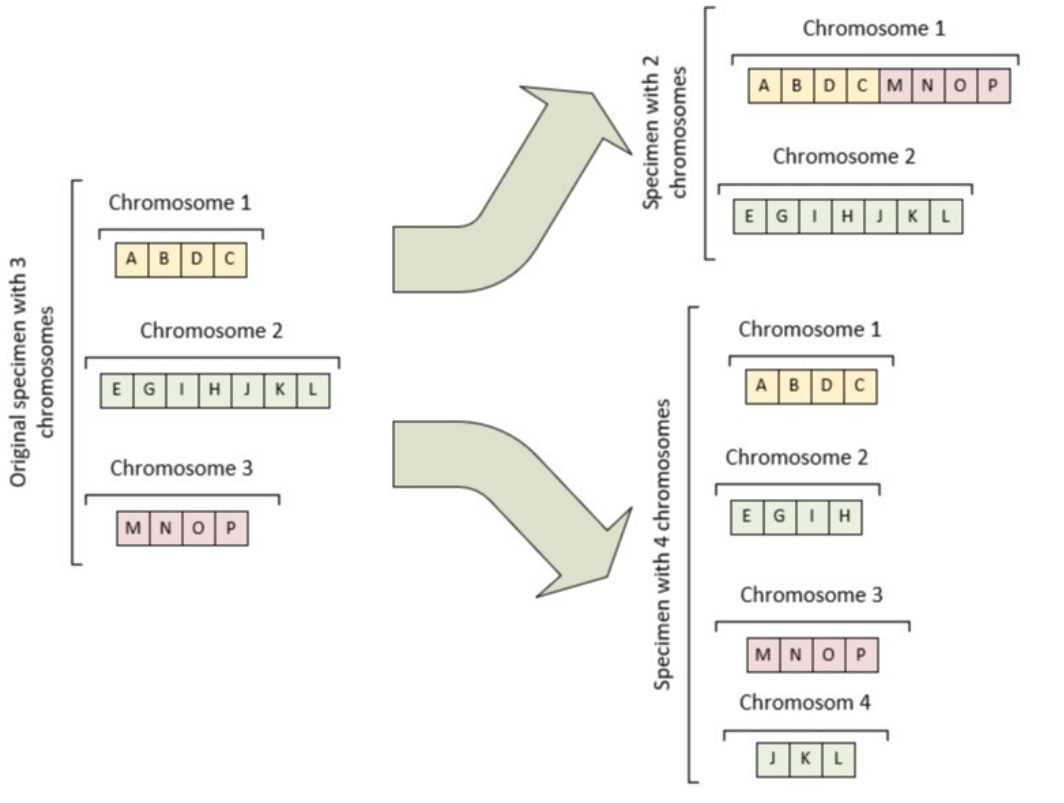


3.3.3.2 Cross-Chromosome Mutators

Cross-Chromosomes will create or destroy chromosomes within the same specimen. This will alter the amount of paths (i.e. minimum required drones) that a specimen has [92] [93] [96]. In Figure 3.11 it can be appreciated how from a single specimen, more may come out. Here the reproduction of a specimen is limited and constrained to the number of chromosomes and points within each

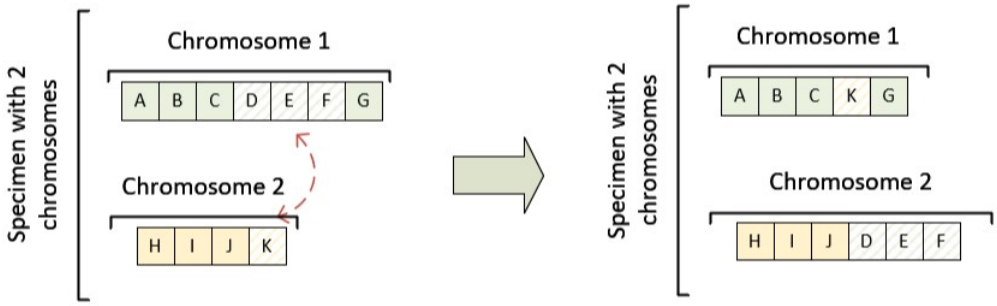
chromosome. This is following a rule that a path may not have less than three points - thus generating a minimal close graph, and how many paths it currently has to apply all mutators.

Figure 3.11: Specimen Creation from an Cross-Chromosome Mutator



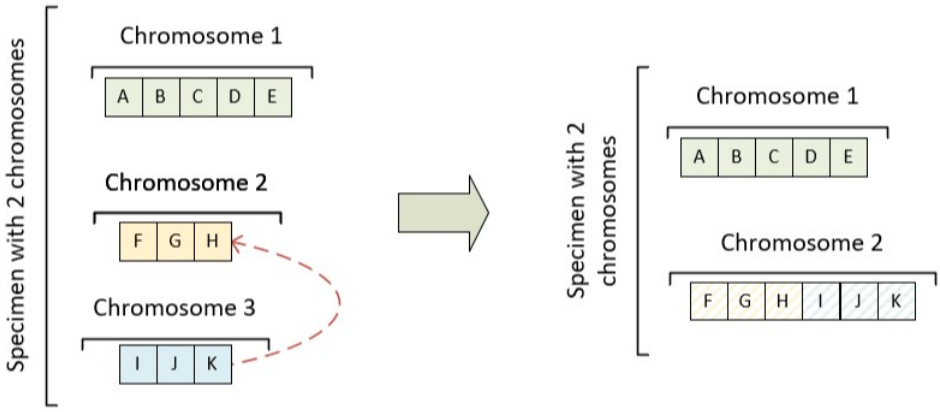
Cross-Chromosome swapping mutator From each of the two selected chromosomes, a section of their chromosomes are selected and swapped. This yields a new specimen with the same amount of chromosomes, but the two chromosomes will have a variation [96]. In Figure 3.12 It can be appreciated how points D, E, F and point K are swapped from chromosome 1 and 2 to chromosome 2 and 1 respectively. This mutator expands the opportunity for a better combination of points between chromosomes, thus resulting in a possible better-fitted specimen.

Figure 3.12: Cross-Chromosome Swapping Mutator



Cross-Chromosome join mutator For two selected chromosomes from a specimen, a third chromosome will be created by joining the selected chromosomes. This will yield a specimen with one less chromosome than the original specimen [96]. In Figure 3.13 it can be appreciated how chromosome 2 and chromosome 3 are joined to form a single chromosome F, G, H, I, J, K. This mutator will allow reduction of chromosomes, thus reducing the number of required drones for the specimen.

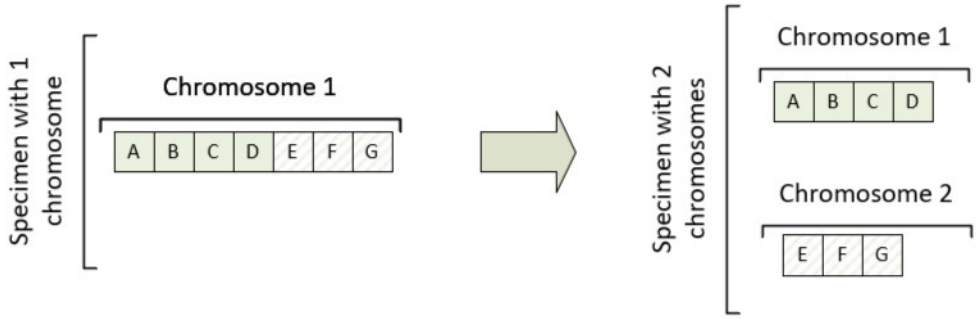
Figure 3.13: Cross-Chromosome Join Mutator



Chromosome separation mutator For a chromosome, a section greater than three points will be selected. That section will be separated, yielding two chromosomes for a new specimen [96]. In Figure 3.14 it can be seen how chromosome 1 is split yielding a new specimen with two chromosomes instead of one. This mutator will allow a gain of specimens that will require more drones,

and probably position themselves as better fitted than the other specimen due to fitness function evaluations.

Figure 3.14: Chromosome Separation Mutator



3.3.4 Fitness Functions

Fitness functions have the task of evaluating a specimen by different parameters and provide an indicator regarding how fit they are given an explicit scenario [92] [93] [96]. For this work, three fitness functions have been implemented. Each one will secure that the selected specimen (i.e., drone path) will be the most efficient, effective, and flyable solution to the proposed problem. The selected weight that each fitness function has over a specimen score is calculated according to (3.17), where SR is the Service Ratio obtained from the service fitness function , AR is the Angle Ratio obtained from the angle fitness function, DR is the distance ratio obtained from the distance fitness function, and IR is the intersections ratio obtained from the intersections fitness function.

$$Score = (0.5 \times SR) + (0.2 \times AR) + (0.15 \times DR) + (0.15 \times IR) \tag{3.17}$$

The described ponderation in (3.17) answer to the best interest for the given scenario and follows a weight and compensation analysis with given a population [96]. By giving a ponderation of 0.5 to SR, we can assure that the specimen with

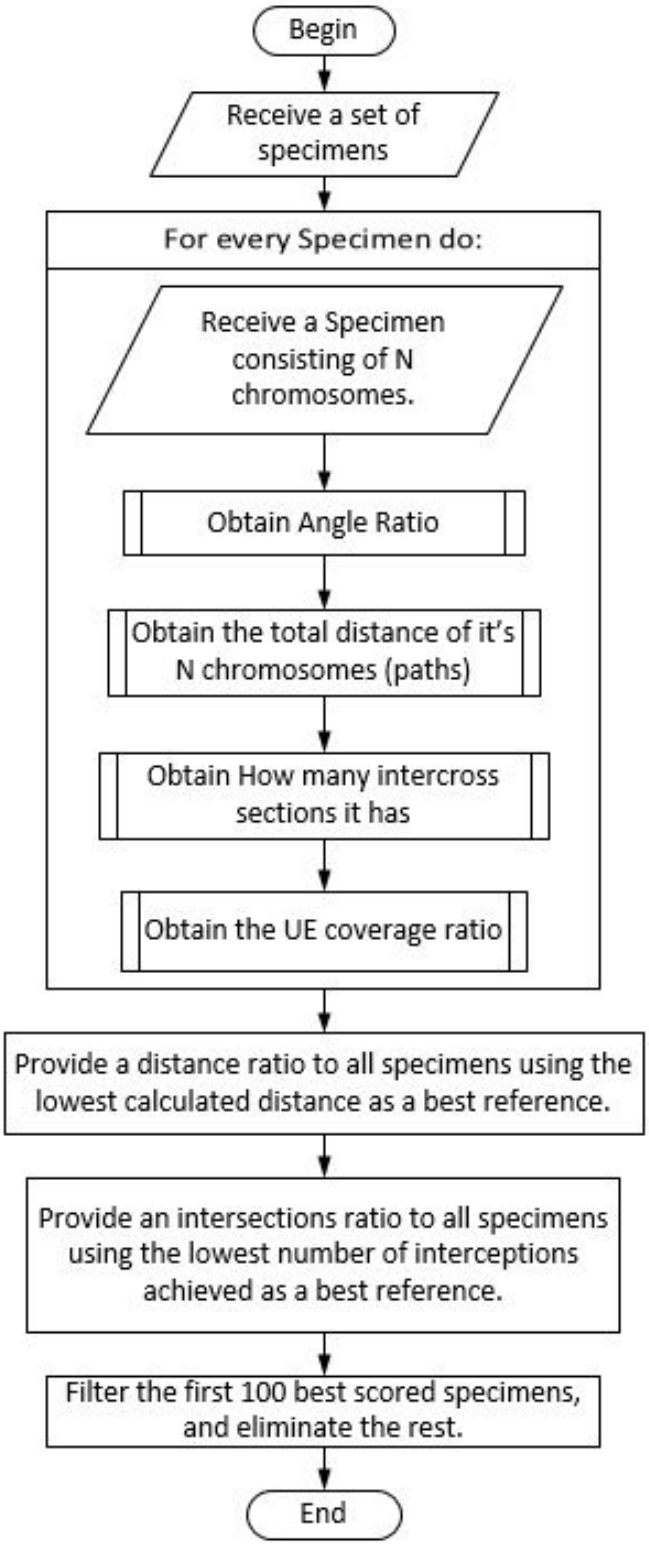
utmost coverage will have a greater probability to keep reproducing. Therefore, if any other specimen may have a better flying solution considering but with lower coverage, the selection process will ensure the specimen with the utmost coverage will be the one to be improved forward. AR have a slight score weigh advantage from DR and IR to ensure that close angles are avoided with the best effort. This will yield specimens that display flyable paths, one of the utmost important parameters for the desired solution. Finally, DR and IR carry an equal, yet similar weight than the other two ratios to provide a finalizing adjustment. After some generations, most specimens within a population will have similar scores regarding SR and AR; it will be here when DR and IR make the difference between the chosen ones, and which should keep reproducing. By doing so, the possibility that a shortest and free from path crossing specimen will be determined will be at its highest. This procedure is applied according to the diagram displayed in Figure 3.15.

3.3.4.1 Offered Service Fitness Function (SR)

This function will calculate how many un-serviced UE have had a connection at one point for all drones travel paths. To perform such calculations (3.1) and (3.3) will be used, with a constant height of 50 meters per drone. To relieve some computer power necessity, the unserviced UE data set is updated as a UE is considered with service, and calculations are done every 3 km within a drone's path.

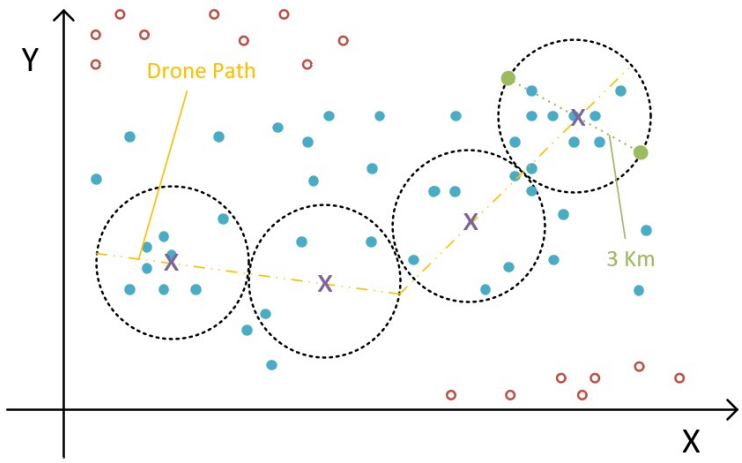
Figure 3.16 refers to a drone path and a 3 km diameter. The 3 km diameter is the distance in all directions by which the next floating point for a MABS will be placed, followed by calculations for UE service coverage (i.e., with the new position, how many new UE will be serviced). For example, in Figure 3.16 the circles with solid filling are UE that have been provided service at one point, while the circles with no filling have not encountered the required threshold

Figure 3.15: Fitness Score Calculation Diagram Flow



coverage service during any drone pathway. The purple X inside he circles refers as to where the MABS will be positioned for calculations. A flow chart of this service function can be found in Figure 3.17.

Figure 3.16: Fitness Service Function Description



3.3.4.2 Angles Fitness Function (AR)

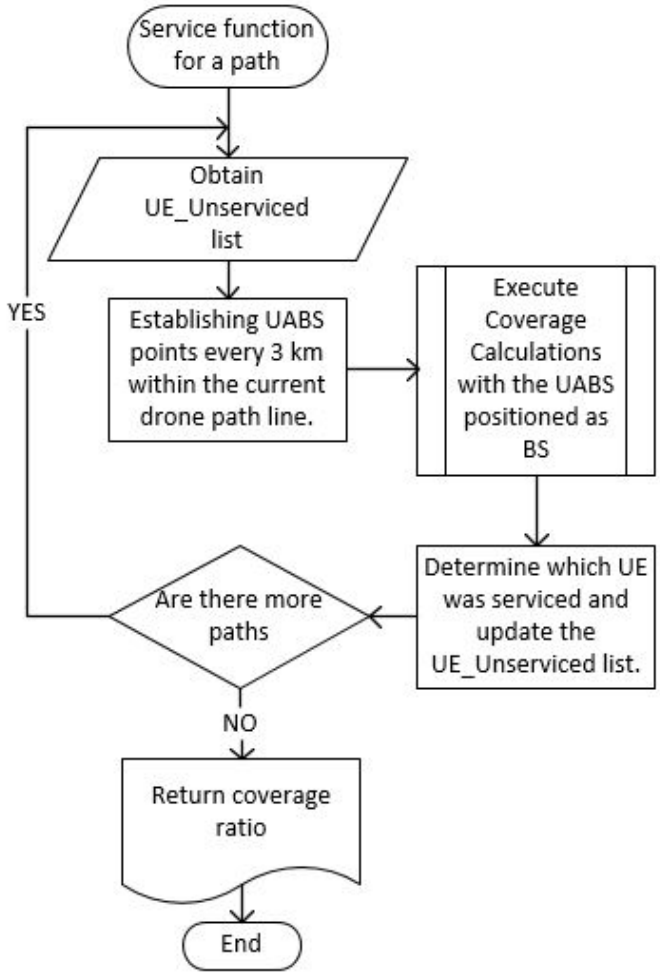
This function will consider a ratio based on (3.18) and (3.19), where n is the total number of angles within a path (i.e. chromosome), and N is the total number of paths within an specimen (i.e. total number of chromosomes).

$$ratio = \frac{|\angle\ (ABC) - \pi|}{\pi} \tag{3.18}$$

$$AR = \frac{\left(1 - \frac{\sum ratio\ \forall\ \angle\ (A,B,C) \in path}{n}\right)\ ,\forall\ path\ ,\in\ specimen}{N} \tag{3.19}$$

By implementing (3.18) and (3.19), the specimen will have a higher score as its' paths flow will have a higher number of obtuse angles. This is because the ratio (single angle ratio) is calculated based on how much it approximates a straight angle (i.e., π for Radians). Furthermore, the average of all angle ratios (AR) is weighted equally so that an acute angle will generate a negative effect on the solution. Even, if there is a triangular formation with a highly obtuse angle,

Figure 3.17: Fitness Service Function Flow Chart



that would yield two very acute angles, resulting in a lower score for that specimen based upon that specific triangular path. In Figure 3.18 we can observe a set of angles, where from left to right the obtained single angle ratio (α) would be in accordance with Table 3.2. Because of this, it is certain that a perfect score for AR cannot be achieved, for it is impossible to establish a close graph with co-linear lines.

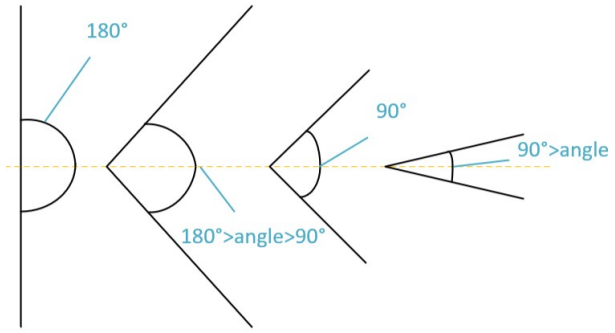
3.3.4.3 Distance Fitness Function (DR)

Because there is no possible method to determine the best distance from the inner perspective of a specimen, a comparative value considering all specimens

Angle	Ratio
$\alpha = 180$	ratio $= 0$
$180 > \alpha > 90$	$0 <$ ratio < 0.5
$\alpha = 90$	ratio $= 0.5$
$\alpha < 90$	ratio < 0.5

Table 3.2: Single Angle Ratio Table

Figure 3.18: Angle Descriptions



was adopted. A distance value (δ) is determined by the (3.20), where E is determined by (3.15).

$$\delta = \sum E, \ \forall \, path, \ \in \ specimen \tag{3.20}$$

After obtaining all distances from all specimens, a minimum distance is selected ($\min(\delta)$). Next, every δ value will be compared to this value, thus obtaining DR, as stipulated in (3.21) where i refers to the DR of a specific specimen.

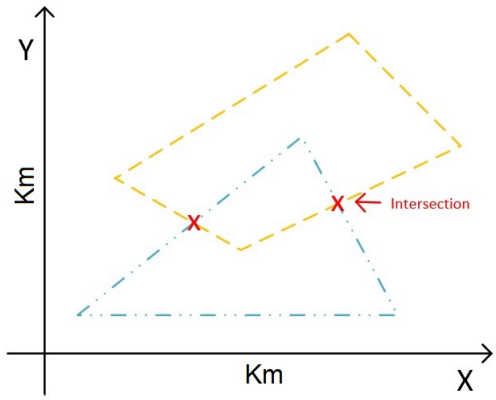
$$DR_i = \frac{\min(\delta)}{\delta_i}, \ \forall \, i \, , \in \ set[\delta] \tag{3.21}$$

While this function will not secure the opportunity for a specimen to be the best candidate for a determined scenario, it will provide a best effort to approach distances in accordance to other specimens, regardless if the specimen with the $\min(\delta)$ is the fittestt.

3.3.4.4 Intersections Fitness Function (IR)

An intersection happens when two line segments meet at a common point. In Figure 3.19 two intersections between two paths are signaled by two X.

Figure 3.19: Intersection Example



Because there is no possible method to determine if a specimen with zero interceptions will yield better results than a specimen with two or more interceptions, another comparative value considering all specimens was adopted.

A number of intersections (Φ) between all paths (chromosomes) is calculated per specimen. After obtaining all Φ, a min(Φ) is determined and used for comparison. In (3.22) we can appreciate how the number of intersections of a specimen is compared to the minimum.

$$IR_k = \frac{\min(\Phi)}{\Phi_k}, \ \forall \ k \ , \in \ set[\Phi] \tag{3.22}$$

Similar to the Distance Fitness Function, this will allow a partial comparison with the lowest Φ without ensuring that the specimen with the lowest Φ is the overall fittest specimen.

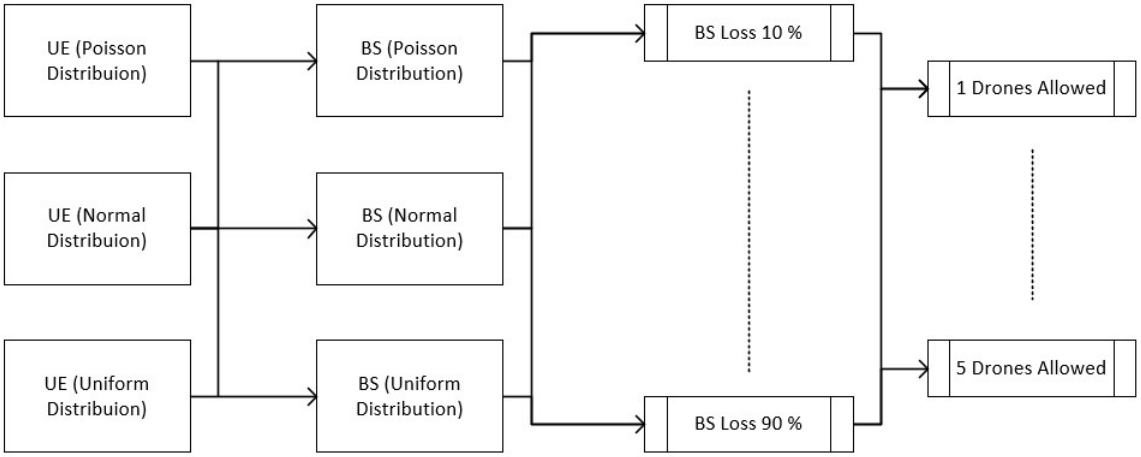
3.4 Simulation Experiment Design

To test and evaluate the proposed system model and algorithm, in the next chapter we will simulate and obtain results to achieve periodic service

communication for UE with no service.

The simulation will consist on a set of experiments that will be executed according to the logical simulation diagram described in Figure 3.20.

Figure 3.20: Logical Simulation Diagram



Furthermore, from the results of the experiments in the next chapter, an analysis will be done considering a comparison between distributions, the effectiveness of the algorithm, and if the goal of providing periodic communication will be achieved.

Chapter 4

Performance Evaluations for Proposed MABS Placement Algorithm

In this chapter, performance results are obtained from the experiments

for the proposed Mobile Aerial Base Station (MABS) deployment schemes last chapter. Specifically, examples of obtained paths are shown, and their evolution explained. Finally, comparative data between different probability distributions are shown to enable analysis and meaning of such figures. These results are detailed and discussed to illustrate the effectiveness of the proposed schemes.

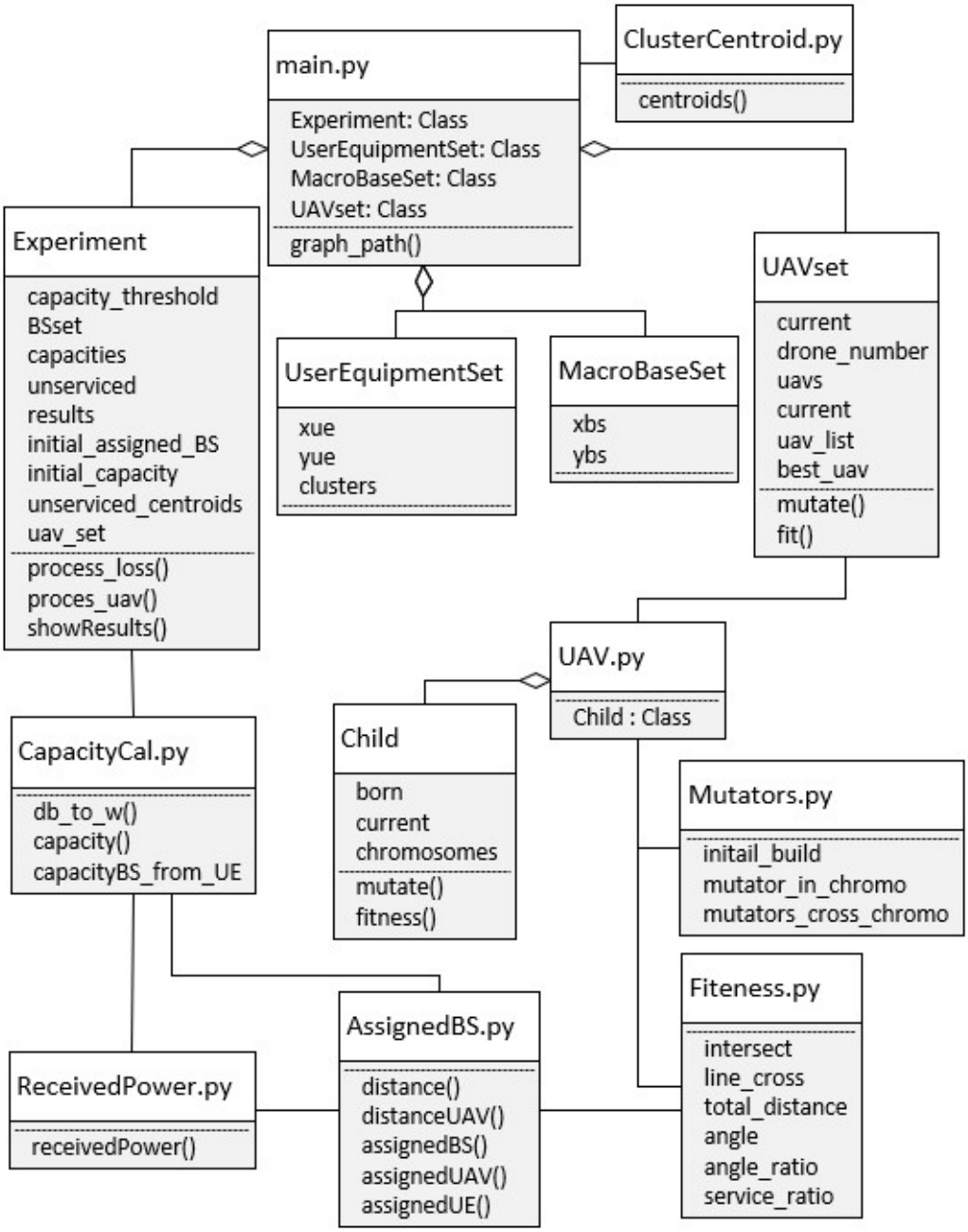
4.1 Software Implementations

In this subsection, we give an overview of how the algorithm is implemented.

The proposed algorithm was implemented using Python version 3.7.1. The documented code can be found in appendix B. A detailed Unified Modeling Language (UML) diagram of the implemented code can be found in Figure 4.1.

The code was separated into different files and classes to be able to perform several experiments in parallel and effectively utilize computer resources. Best coding practices were applied to ensure computational resources optimization and thus achieving results in effective time.

Figure 4.1: UML Diagram of Implemented Code



4.1.1 Algorithm Considerations

Within the code and GA algorithm logic, for the best specimen to be determined for each experiment iteration, a candidate will be selected according to its score. Each iteration is considered to be a new generation. For each new generation, selected specimens will be mutated and scored as detailed in Chapter 3.

A population of 100 will be used for this experiment for the reproduction pool of the genetic algorithm. This will ensure diversity and allow the evolution of multiple specimens regardless if they are currently the best specimen. This is done to ensure that a single solution is not over-breaded, thus limiting the number of possible solutions to a variation of the first best specimen that achieved an outstanding score [96].

A cycle lock of 40 generations was used for each experiment. Therefore, after the best specimen is selected, if in 40 new generations a new specimen cannot beat the current prime specimen, then that specimen will be selected as the best specimen for that iteration. This number of generations provides an adequate level of stabilization for a single best solution to prominently outstand other specimens based on the score provided by the fitness functions [96].

Given how the algorithm works, the execution time grows linearly according to the number of calculations required, provided an increment of UE or BS regardless of the distribution. This presents an algorithm that behaves at a equivalent of an $O(n)$ notation [97] [98].

Finally, to ensure a good sampling for the probability distribution results, each figure provides data calculated from an average of 100 simulation results combined.

4.1.2 Number and Positioning of UE and BS

Positioning was done by following a uniform distribution for every point, where a value between 1 and 25000 was chosen for an axis. Therefore, every UE and BS would have two random values, where the first one belongs to the X-axis and the second one to the Y-axis. This value would be represented in meters.

The numbers of UE and BS were obtained by following three distributions.

4.1.2.1 Poisson Distribution

The number of UE and BS to be deployed is calculated by employing a λ as described in (4.1) [99].

$$\lambda = multiplier \ \times \ area \tag{4.1}$$

This distribution is required to yield a number very close to 2500 and 250 for UE and BS respectively, thus a multiplier of 100 and 10 will be used respectively.

4.1.2.2 Normal Distribution

For a normal distribution, a Gaussian bell was used with a mean of 100 and a Standard Deviation (σ) of 10 for the number of UE. For the number of BS the mean will be 10 with a σ of 1. This would allow a number to be around 2500 and 250 for UE and BS receptively. In addition, the selection of σ for both cases would allow a good representation of a required number, but still with a variation [99].

4.1.2.3 Uniform Distribution

For a Uniform distribution, a number between 1 and 2500 will be selected for the number of UE, where every number has the same probability. As for the number of BS, the number will be between 1 and 250.

4.2 Simulation Results

In this section, direct results from different experiments are shown. A total of 405 (3 UE distributions x 3 BS distributions x 9 loss levels x 5 allowed drones) best specimens were obtained after simulations.

Acronym	Description
UBPP	User Equipment number obtained with Poisson Distribution and Base Station number obtained with Poisson Distribution
UBPN	User Equipment number obtained with Poisson Distribution and Base Station number obtained with Normal Distribution
UBPU	User Equipment number obtained with Poisson Distribution and Base Station number obtained with Uniform Distribution
UBNP	User Equipment number obtained with Normal Distribution and Base Station number obtained with Poisson Distribution
UBNN	User Equipment number obtained with Normal Distribution and Base Station number obtained with Normal Distribution
UBNU	User Equipment number obtained with Normal Distribution and Base Station number obtained with Uniform Distribution
UBUP	User Equipment number obtained with Uniform Distribution and Base Station number obtained with Poisson Distribution
UBUN	User Equipment number obtained with Uniform Distribution and Base Station number obtained with Normal Distribution
UBUU	User Equipment number obtained with Uniform Distribution and Base Station number obtained with Uniform Distribution

Table 4.1: Acronym Descriptions

Because of the extensive amount of data handled and collected over multiple experiments, only a variated sampling of the obtained results will be shown (For distribution comparison results, all results will be taken into consideration).

Table 4.1 shows the description of the acronyms used in the rest of this thesis.

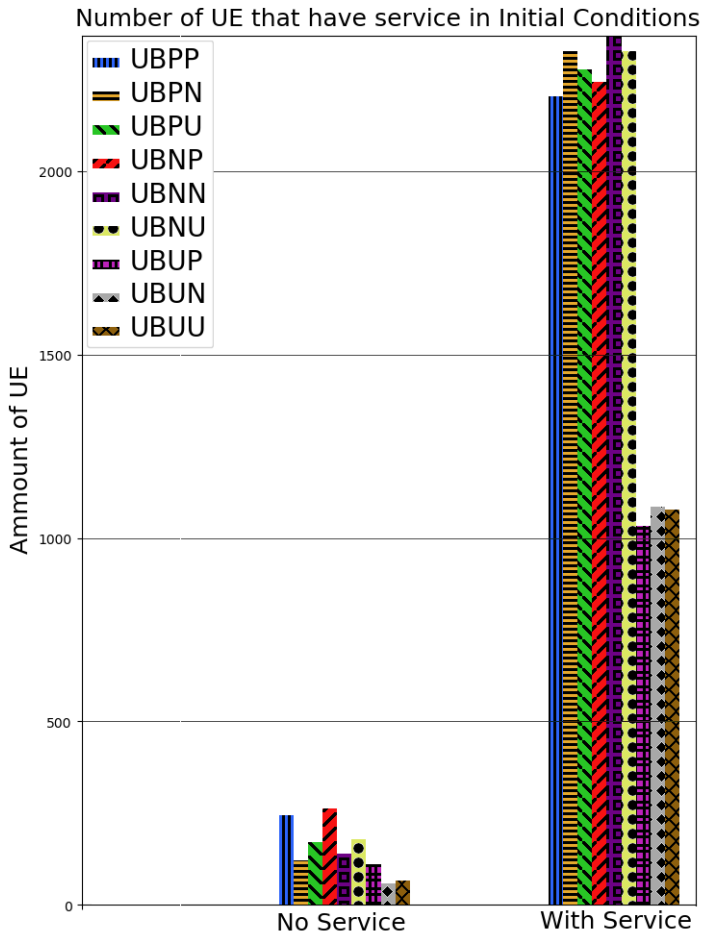
In addition, the term *Have Service* that refers to a UE is to be understood as a set of UE having to receive a throughput capacity equal to or greater than 200 Kbps. Moreover, a MABS would be considered as a Drone during the result analysis.

4.2.1 Initial Scenario

In this section, we analyze some aspects and considerations that will allow a better understanding of future results presented in this thesis.

An initial analysis of how the scenario is presented with no alterations is shown in Figure 4.2.

Figure 4.2: Initial Service for all Distributions



An immediate analysis of Figure 4.2 can provide information about a great difference between UE with service and with no service. The number of UE with no service is below 200, except for UBPP and UBNP where the number rounds 250. While for UE with service, UBPP, UBPN, UBPU, UBNP, UBNN, and UBNU have UE with service above 2000; in addition, the number of users will be similar, yet due to differences regarding density and positioning, small differences

regarding a number of serviced users can be found.

It may also be noted how in UBUU, UBUN, and UBUP have a considerably lower number of users. This is solely due to the density value considered for the experiment. Moreover, given that the number of UE is low, and that UBUP and UBUN still have a high number of BS, provides a much lower number of unserviced UE.

The relation provided by the different distribution should only be analyzed from a distribution difference perspective, and not empirical. This is because the goal of using different behaviors is to determine if there's a difference in behavior during path creation experiments.

4.2.1.1 Voronoi Graphs

A usual representation for a UE and BS deployment for a cellular service coverage perspective can be obtained from displaying the established points within a Voronoi diagram. In the following Voronoi figures, UE and BS are represented by circles and triangles respectably.

UBPP and UBPN Voronoi diagrams are shown in Figures 4.3 and 4.4 respectively. It can be observed the similarities between both given the configuration described in 4.1.

However, in Figure 4.5, while the figure looks similar, for this case it would just represent a causality, due to the nature of the Uniform Distribution. However, it is noticeable to some point how the BS density is lower, thus yielding bigger cells for a visual representation.

In Figure 4.6 we can observe UBNN Voronoi diagram. The representation carries many similarities with a UBPP given that the probability distributions provide a density number close to the desired one.

In Figure 4.7 we can observe how the density of UE and BS is lower than other figures. Thus Voronoi with bigger cells is presented in Figure 4.7.

Figure 4.3: Voronoi Diagram for UBPP

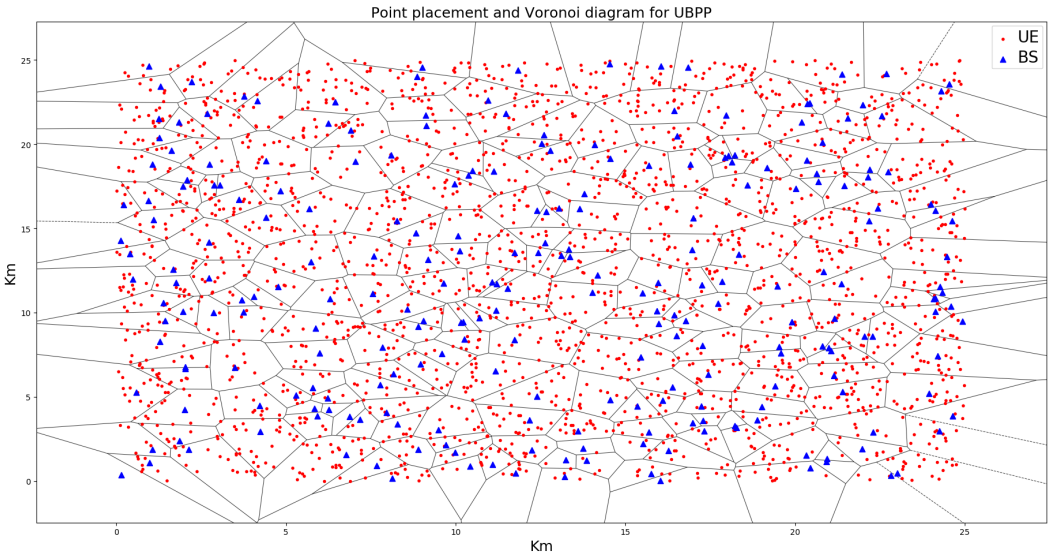
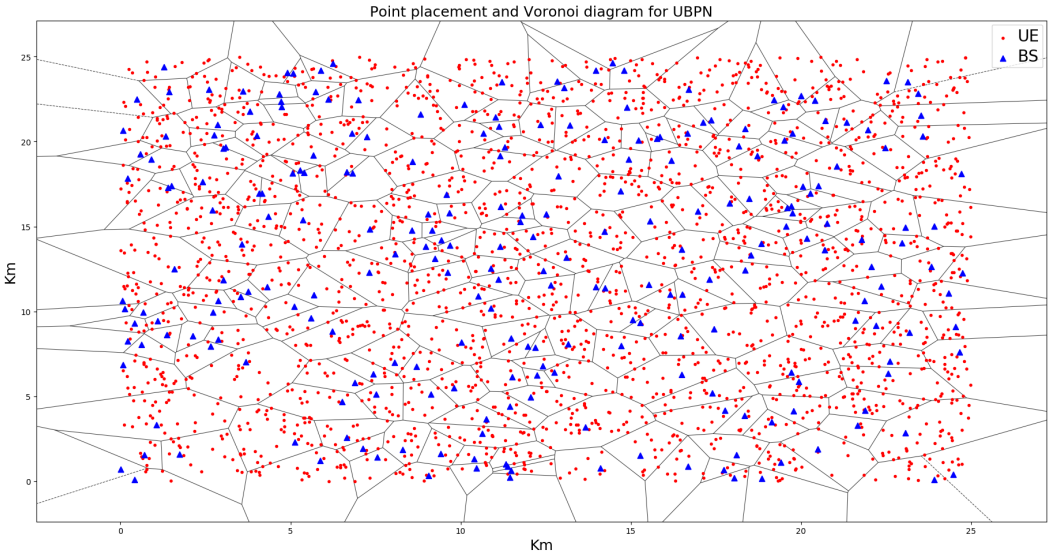


Figure 4.4: Voronoi Diagram for UBPN



4.2.2 Service Loss

Loss of service can be interpreted as a user who has lost its capacity to exchange data with a BS. For this case, this may happen if a UE is out of coverage, or is positioned in a high-density UE area. Thus a UE will no longer be able to achieve the threshold of 200 Kbps minimum considered for this set of experiments.

Loss of service was simulated by erasing a percentage of BS randomly. By doing so, a set of UE for a certain amount of loss will be created, thus leaving

Figure 4.5: Voronoi Diagram for UBPU

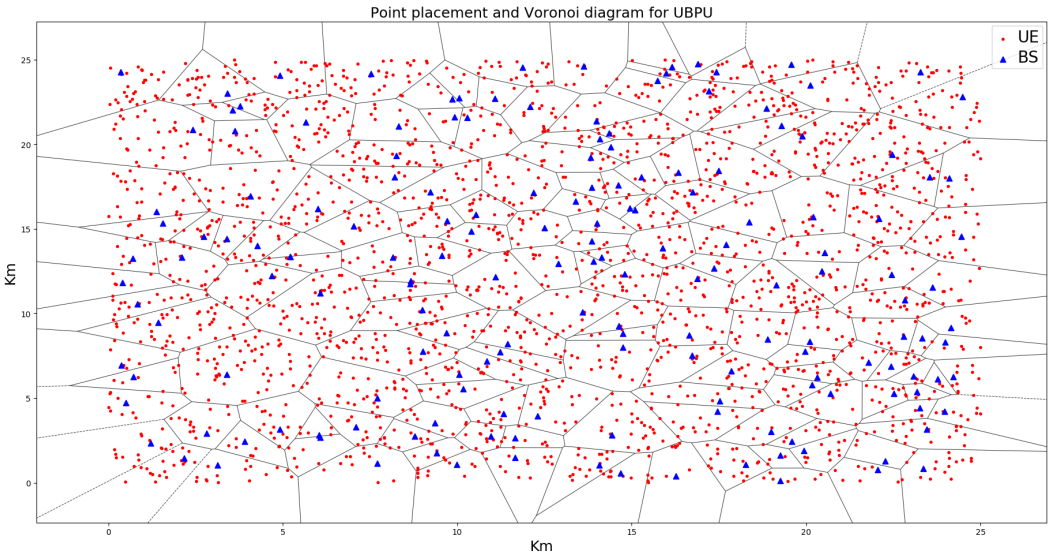
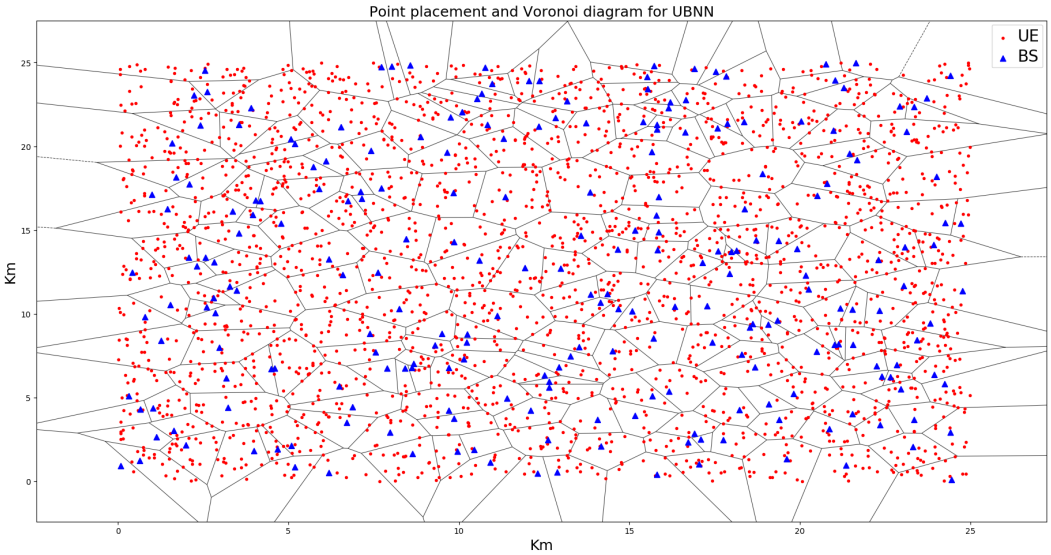


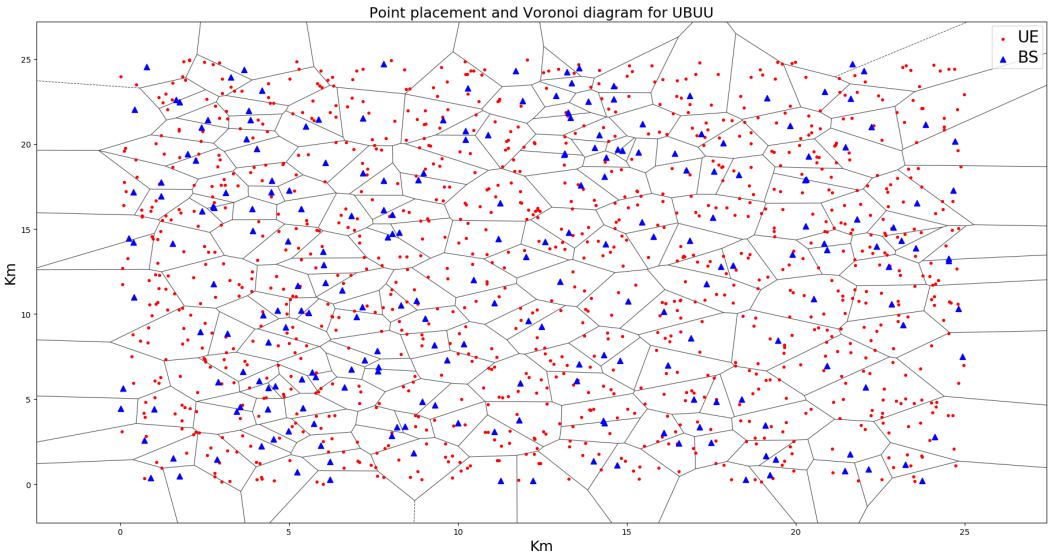
Figure 4.6: Voronoi Diagram for UBNN



several UE with no service. This was implemented as previously formulated in Figure 3.20, where a range of percentage losses from 10% to 90% will be considered for the simulations.

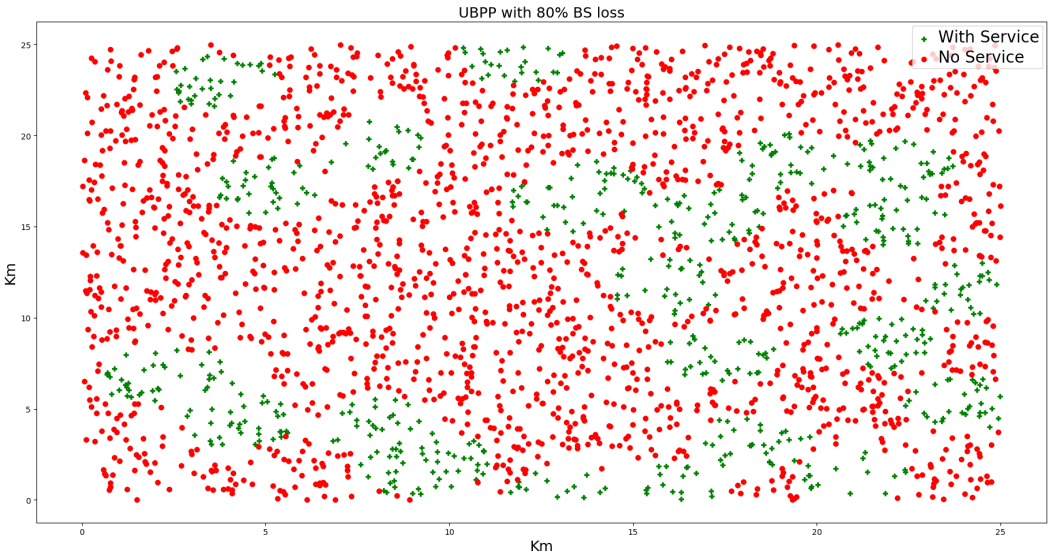
Figure 4.8 depicts a visual representation of a scenario with 80% of BS loss. A radius of x can be found around areas where a BS was not destroyed. It is important to consider that these areas may be understood to be with service coverage, whether they are still being serviced by the original BS or by a rescue

Figure 4.7: Voronoi Diagram for UBUU



team service BS.

Figure 4.8: 80% BS Loss in UBPP



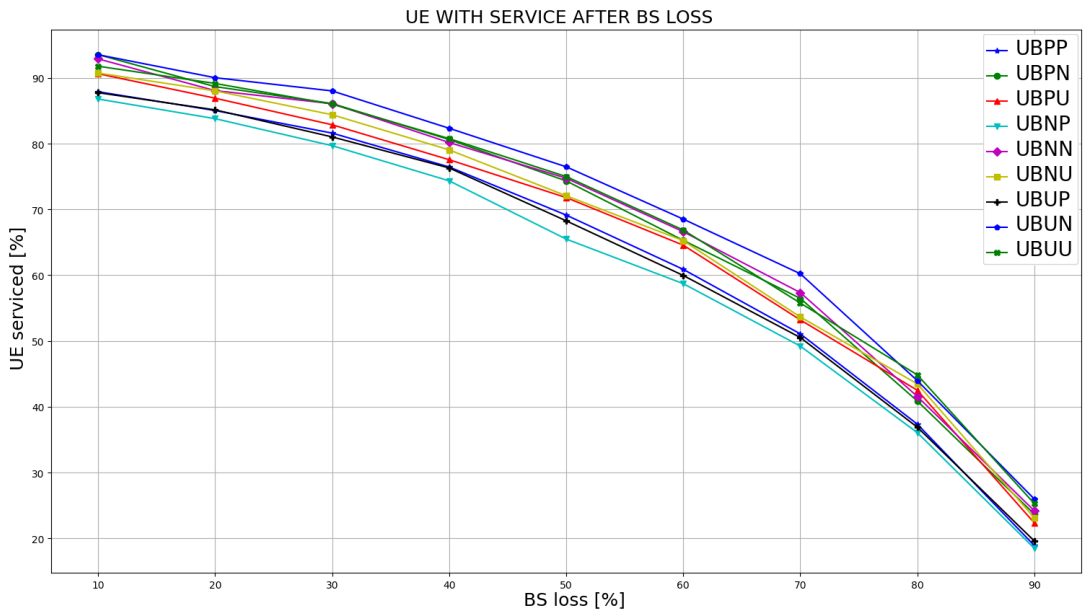
In Figure 4.9 a set of curves provide information about the percentage of UE without service after BS loss. As stated in Figure 4.2, the initial was not complete, and with 10% of BS loss, service is not heavily compromised yet. This is because of the high BS density considered at the initial condition. However, as BS loss increases, a downward curve with a concave arch towards the origin is shown for every distribution. This depicts that the greater the BS loss, a steeper

increase of service for UE is found. Being so, all distributions present only a 25% of UE with service given a 90% of BS loss.

In addition, UBPP and UBUP present a degrading behavior harmonious with the other curves, while still presenting a lower amount of offered service. In addition to UBNP, the number of UE for Poison Distribution appears to be much higher than what the present BS can offer in comparison to other distributions.

As for other distributions, the best-presented case can be seen in for UBUN in almost all loss percentages and is always surpassed by UBUU for an 80% BS loss. This may be due to a lower number of UE, thus the need to provide a service to a lower number of UE.

Figure 4.9: UE Percentage Service Deficiency After BS Loss



4.2.3 Visual Path Examples

In this subsection, the GA algorithm has been executed, and the best specimens have been obtained. Some of these specimens will be described and shown to provide a visual representation of the applied algorithm in the scenario.

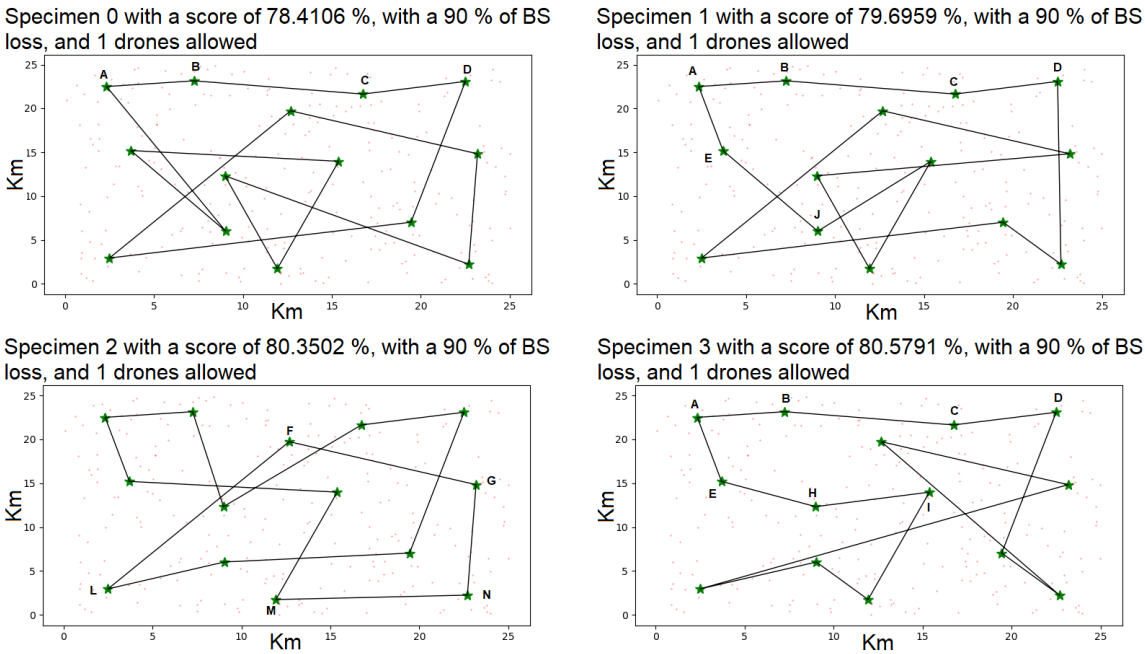
For the following figures, the name specimen will be used to describe a selected best specimen and not the specimen number used for the simulations.

In Figures 4.10, 4.11, and 4.12 we can observe an example of a best specimen evolution for UBUU with 90% of Base Station loss, one drone allowed.

In Figure 4.10 the first best specimen (specimen 0) to be selected is one with a straight flying path conformed by the path $A \rightarrow B \rightarrow C \rightarrow D$. All other centroids appear to scramble and showing obtuse angles, which are not desired unless otherwise. For specimen 1, a better path comes up after applying mutation functions for some generations. Now a path conformed of $J \rightarrow E \rightarrow A \rightarrow B \rightarrow C \rightarrow D$ provides a better flying solution. However, the flight path does not include all the points, and there are still paths that create unwanted flying patterns. It is also noticeable how specimen 2 came from a mutation from specimen 1. This is not the case for all specimens since regardless of the best-selected specimen, there's still a population of 100 more specimens being mutated and selected for future reproduction. For specimen 2, it's noticeable how it's not a direct decadent from specimen 1 nor 2. This is because it lacks the straight path $A \rightarrow B \rightarrow C \rightarrow D$. However, the best selection came to be by covering a greater distance with an acceptable flying path even though it involved fewer visited centroids. This path is conformed by $L \rightarrow F \rightarrow G \rightarrow N \rightarrow M$. In addition, it provides more straight paths between other centroids; yet, these paths are still not optimized to be the best flyable path. In specimen 3 another descendant from specimen 1 may be found. This time, the specimen presents a path $I \rightarrow H \rightarrow E \rightarrow A \rightarrow B \rightarrow C \rightarrow D$. The specimen still needs to consider the rest of the centroids and provide an enhanced solution accordingly.

Specimen 4 in Figure 4.11 appears to be a descendant of specimen 1 given their top and lower straight path similarities. This specimen presents a path $G \rightarrow N \rightarrow M \rightarrow J \rightarrow L \rightarrow E \rightarrow A \rightarrow H \rightarrow B \rightarrow C \rightarrow D$. This paths provides

Figure 4.10: UBUU with 90% Base Station Loss and One MABS Allowed, Part 1



an open figure with less intersections and fewer acute angles.

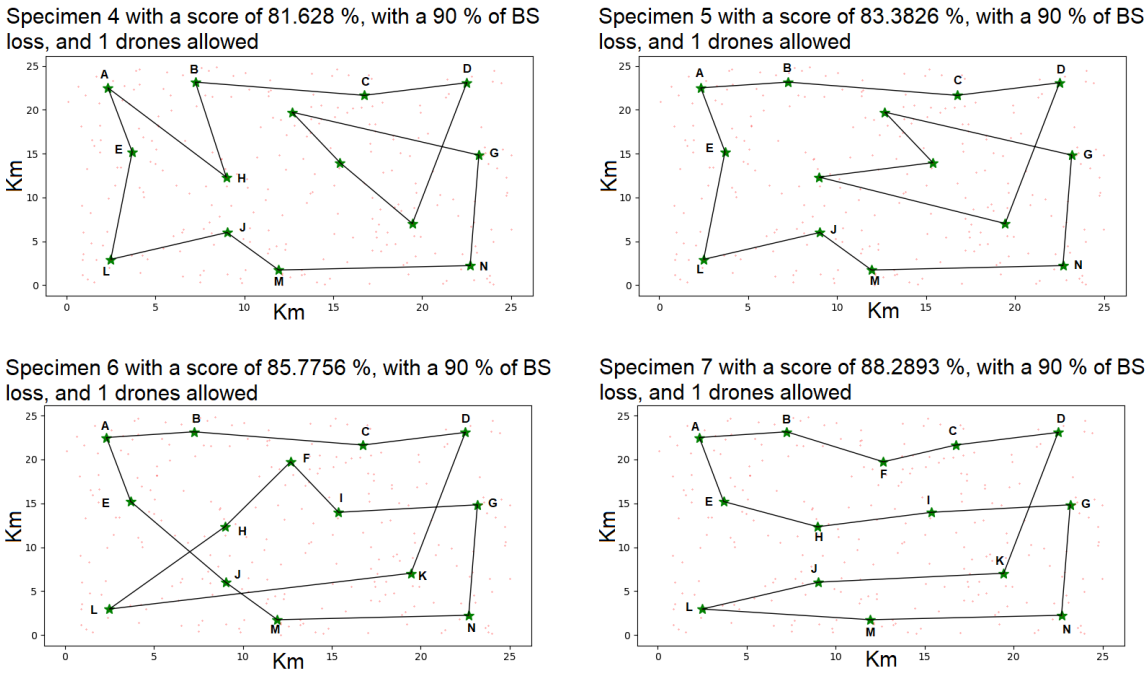
Specimen 5 and specimen 4 appear to be from a common descendant. However, specimen 5 presents a bigger flyable path comprised of $G \to N \to M \to J \to L \to E \to A \to B \to C \to D$, thus achieving a higher score.

Specimen 6 presents a variation where it excludes node L from the greater outside path and includes it in its' inner path.

Specimen 7 appears to not have any immediate common descendant from specimen 4 due to how it's paths are determined. This is proof that best specimens are selected from a variated pool of candidates in competition, and not the same candidate mutating over and over again. It presents an acceptable flyable path $G \to I \to H \to E \to A \to B \to F \to C \to D$. It creates a long break in the path to reach K, in addition to a rough 90-degree break in angle $\angle NGI$.

Finally, in Figure 4.12 the best candidate is shown. This candidate came to be after not been replaced by a better one for 40 consecutive generations. It

Figure 4.11: UBUU with 90% Base Station Loss and One MABS Allowed, Part 2

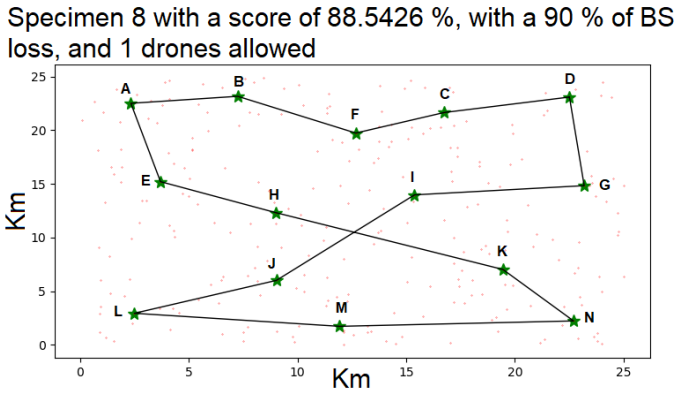


shows a path

$$K \to N \to M \to L \to J \to I \to G \to D \to C \to F \to B \to A \to E \to H \to K.$$

While it does present an acute angle $\angle JLM$, by visual analysis, it may be the only solution viable considering that if the angle were $\angle ELM$, there would still be a need to include centroids J and H.

Figure 4.12: UBUU with 90% Base Station Loss and One MABS Allowed, Part 3



The specimen also has an intersection of its path in the middle, yet, this intersection allows a better flying path than if it were a complete open graph.

Specimen 8 was selected with a final score of 88.5426% and complied with fly-ability to its' best capacity. In addition, the solution also provides coverage to all desired centroids, thus not being a need for a path $L \to E$ to provide service to the UE not near any cluster centroid.

In Figure 4.13 we can observe the evolution of a specimen for a UBUU in a scenario with 30% of BS loss, and a maximum of 4 drones allowed.

Specimen 0 shows three paths, $C \to A \to E \to C$, $I \to B \to F \to I$, and $H \to G \to D \to H$. It can be seen how there are eight intersections between the paths. Although the specimen provides 100% service coverage, it is not the most optimum solution available.

In specimen 1, three new paths are created, $E \to C \to B \to E$, $H \to G \to A \to H$, and $I \to F \to D \to I$. This solution presents a great inconvenience, the path $I \to F \to D \to I$ is not an adequate path, and intersects another path.

Specimen 2 presents 2 paths, $E \to C \to B \to E$ and $I \to H \to G \to D \to A \to F \to I$. This solution reached a score of 86.3971% where a larger path is isolated to ensure the best use of available resources given a maximum of 4 drones.

In Figures 4.14,4.15,4.16,4.17,4.18,and 4.19 we can appreciate a UBPP specimen for a scenario with 80% service loss and 1 drone allowed.

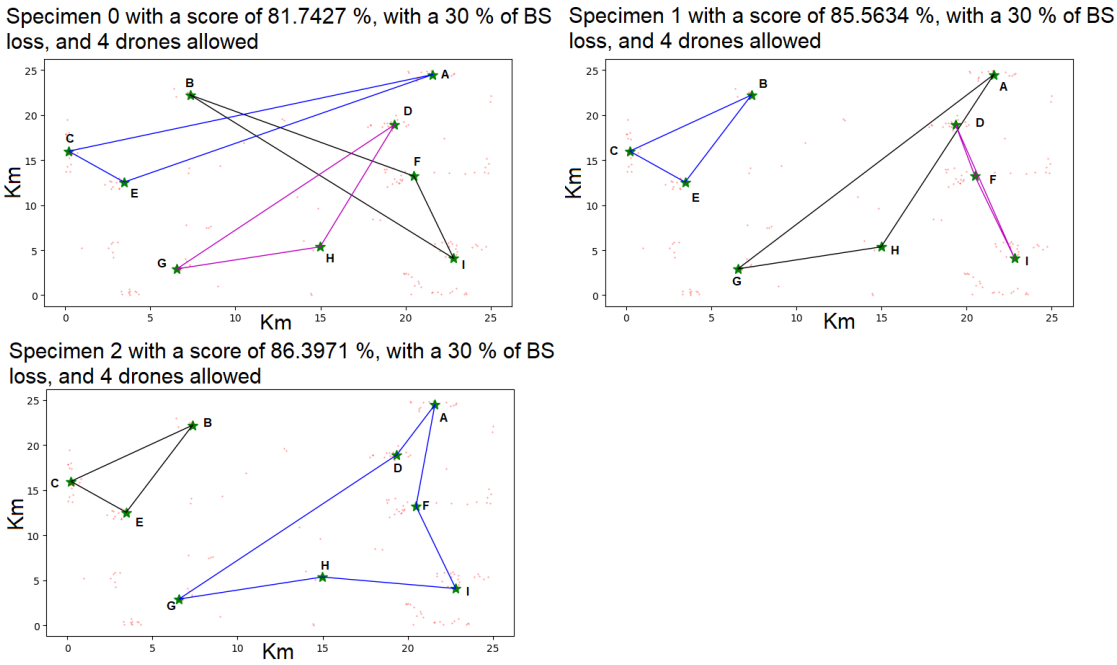
Specimen 0 in Figure 4.14 shows a figure where all paths appear to be scrambled. This specimen gets a score of 73.99% given that it provides enough service coverage and the distance is lower than its contenders.

Specimen 1 keeps presenting a randomized graph at first sight, yet it achieved a score of 75.87% by achieving better distances and angles.

Specimen 2 aside from showing a better performance concerning angles, it shows a clear flyable path section $A \to B \to C \to D$.

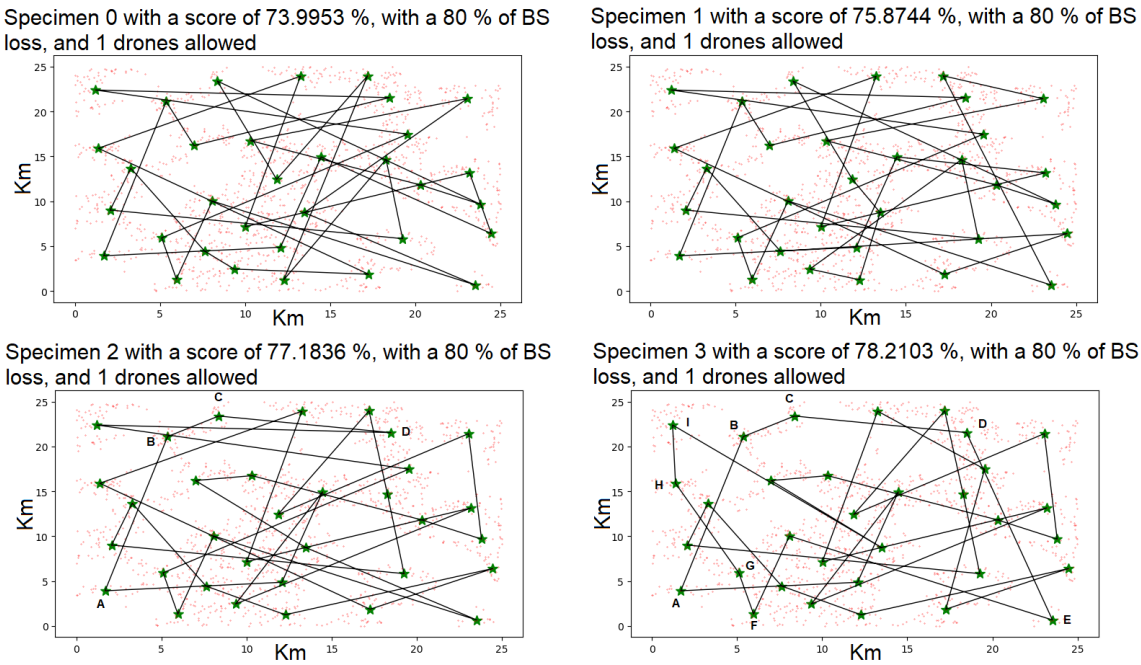
Specimen 3 is clearly a descendant of specimen 2 many shape similarities. It

Figure 4.13: UBUU with 80% BS Loss and 4 MABS Allowed



shows two flyable path sections, $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E$ and $F \rightarrow G \rightarrow H \rightarrow I$.

Figure 4.14: UBPP with 80% BS Loss and 4 MABS Allowed, Part 1



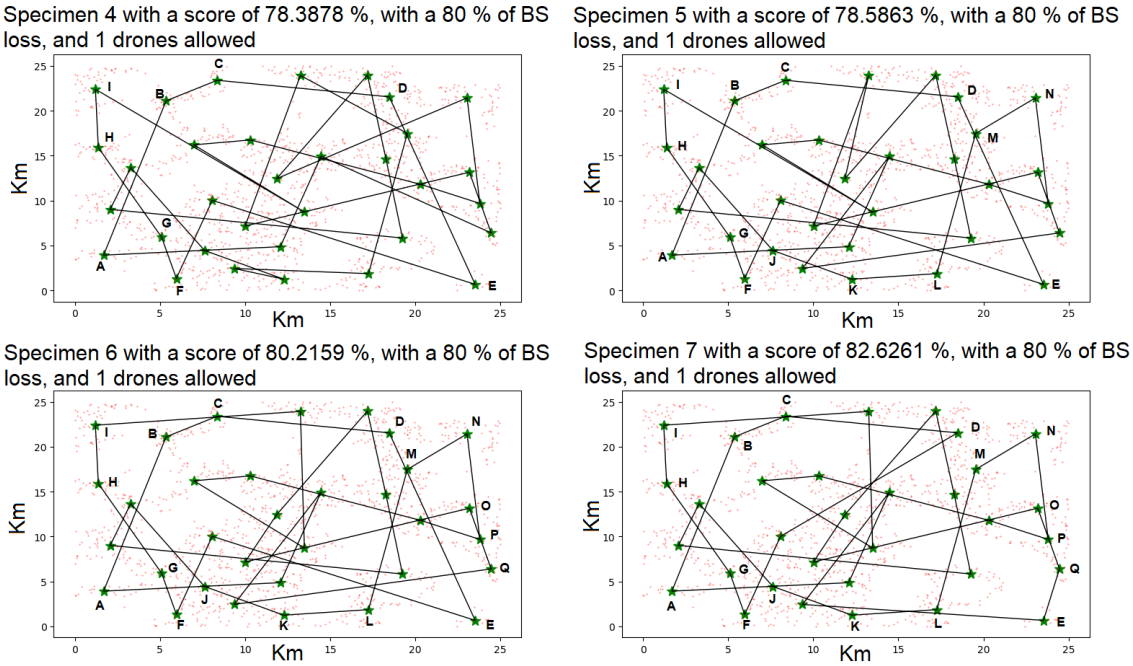
Specimen 4 in Figure 4.15 is a descendant of specimen 3. It presents the same located flyable paths and better performance considering angles between

some centroids.

Specimen 5 is a descendant of specimen 4. It shows three flyable paths, $N \to M \to L \to K \to J$, $A \to B \to C \to D \to E$, and $F \to G \to H \to I$.

Specimen 6 and specimen 7 follows previous specimen evolution lineage, and it starts to show open graphs with better angles and fewer intersections between path segments.

Figure 4.15: UBPP with 80% BS Loss and 4 MABS Allowed, Part 2

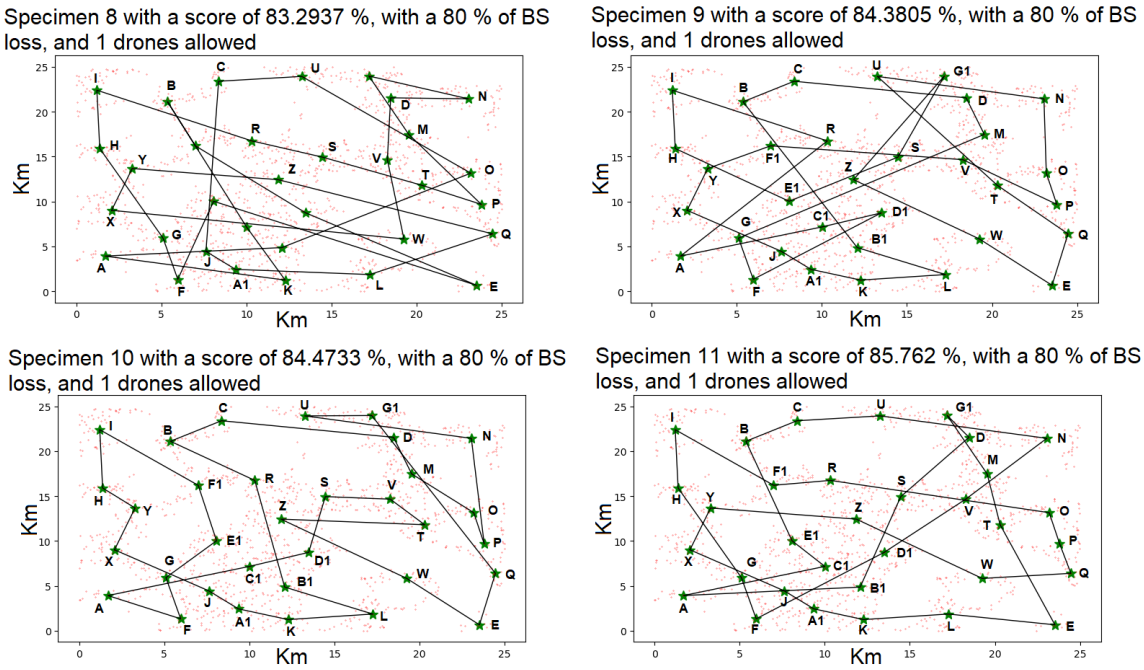


Specimen 8 in Figure 4.16 is not a direct descendant of direct lineage from specimen 4. This specimen was evolving within the provided population of 100 and became the best specimen with a score of 83.2937%.

Specimen 9 and 10 are descendant of specimen 8. In both, an open graph starts to be more noticeable, thus achieving better scores by achieving the same service coverage with lower distances and the same angles.

Specimen 11, not following the lineage of specimen 10 shows long flyable paths. Some clear ones are $E \to L \to K \to A1 \to J \to G \to X$, $B \to C \to U \to N$, and $I \to F1 \to R \to S \to V \to O \to P \to Q$.

Figure 4.16: UBPP with 80% BS Loss and 4 MABS Allowed, Part 3



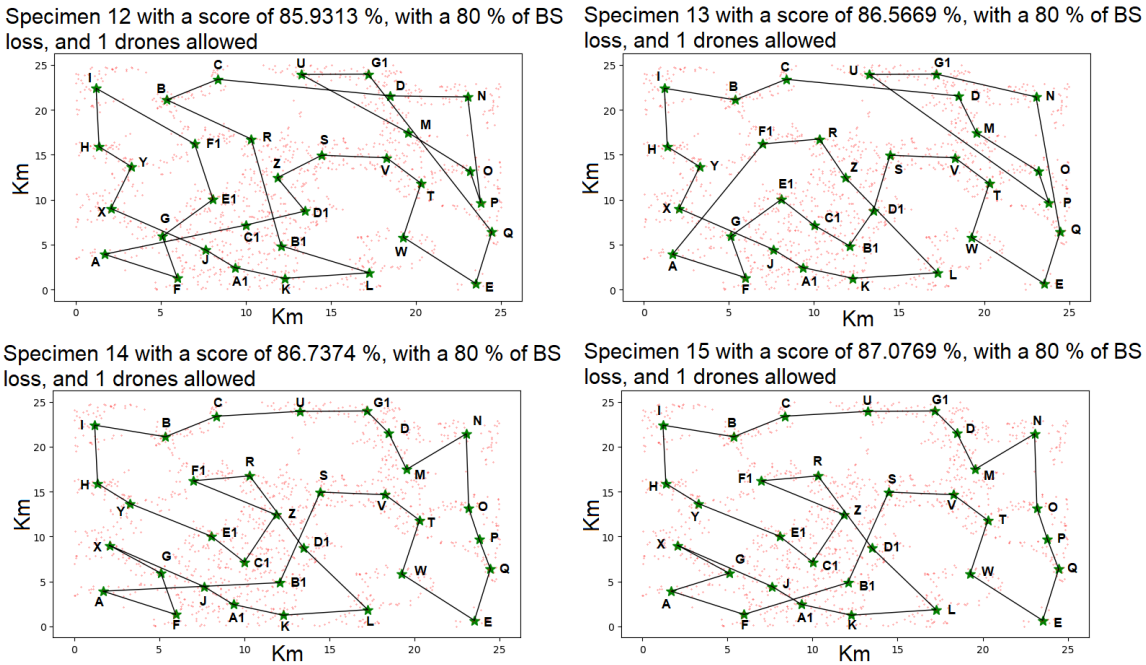
Specimen 12 in Figure 4.17 is a descendant of the lineage of specimen 10 again. This can be noted by following the path segment $E1 \rightarrow F1 \rightarrow I \rightarrow H \rightarrow Y \rightarrow X$.

Specimen 13 a descendant of specimen 12, presents an outer path $Y \rightarrow H \rightarrow I \rightarrow B \rightarrow C$. In specimen 14 the path segment $U \rightarrow G1 \rightarrow D \rightarrow M \rightarrow N \rightarrow O \rightarrow P \rightarrow Q \rightarrow E \rightarrow W$ is attached to the previous segment, thus creating a larger outer path. Specimen 15 is a variation with better angles, thus achieving a score of 87.0769%.

Specimen 16 and 17 in Figure 4.18 begins to form a shape of an open graph; thus fewer intersections and closed angles are achieved. These specimens are descendants of specimen 15.

Specimen 18 is not a direct descendant from specimen 17. It still shows a similar outer path segment. However, the internal path segments comprised of centroids $F1, RE1, C1, Z, B1, S, V, J$ and G form completely different patterns from the ones seen in previous specimens; thus presenting as a new contender.

Figure 4.17: UBPP with 80% BS Loss and 4 MABS Allowed, Part 4



This specimen has a score of 87.9118%.

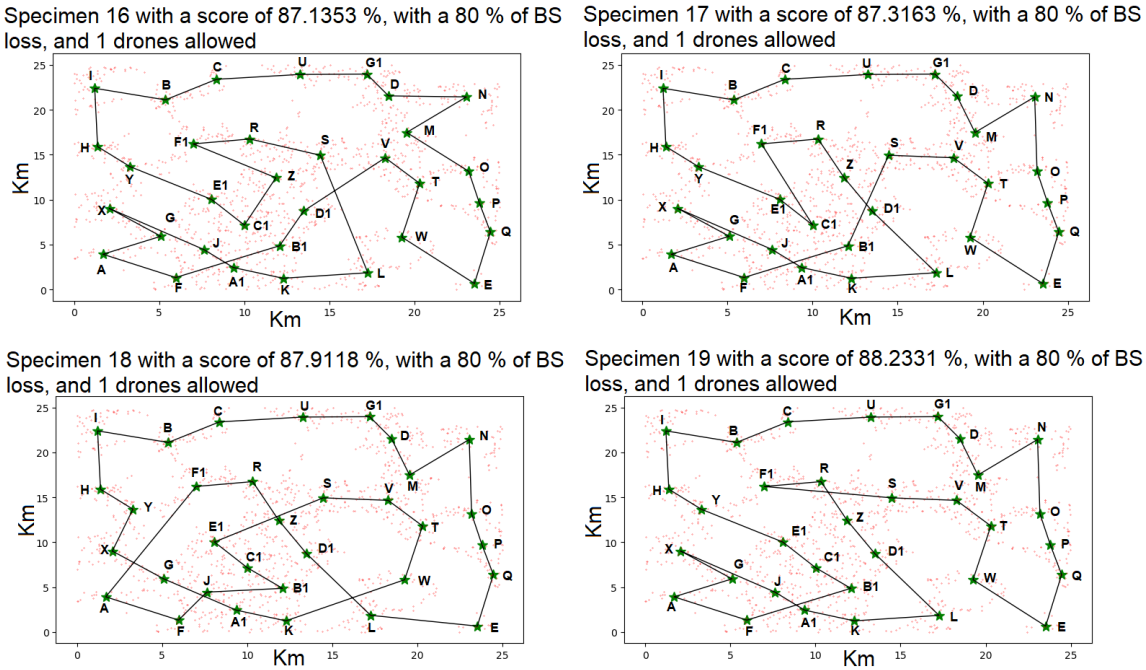
However, specimen 19 becomes the best specimen with a score of 88.2331% being a descendant of 17. This specimen shows improvements from its descendants by providing path segments with more obtuse angles.

Specimen 20 in Figure 4.19 shows greater improvement in reducing the total distance, while maintaining obtuse angles and service coverage.

Specimens 21 and 22 provide some adjustments that enable better flying conditions by forcing some loops within some path segments.

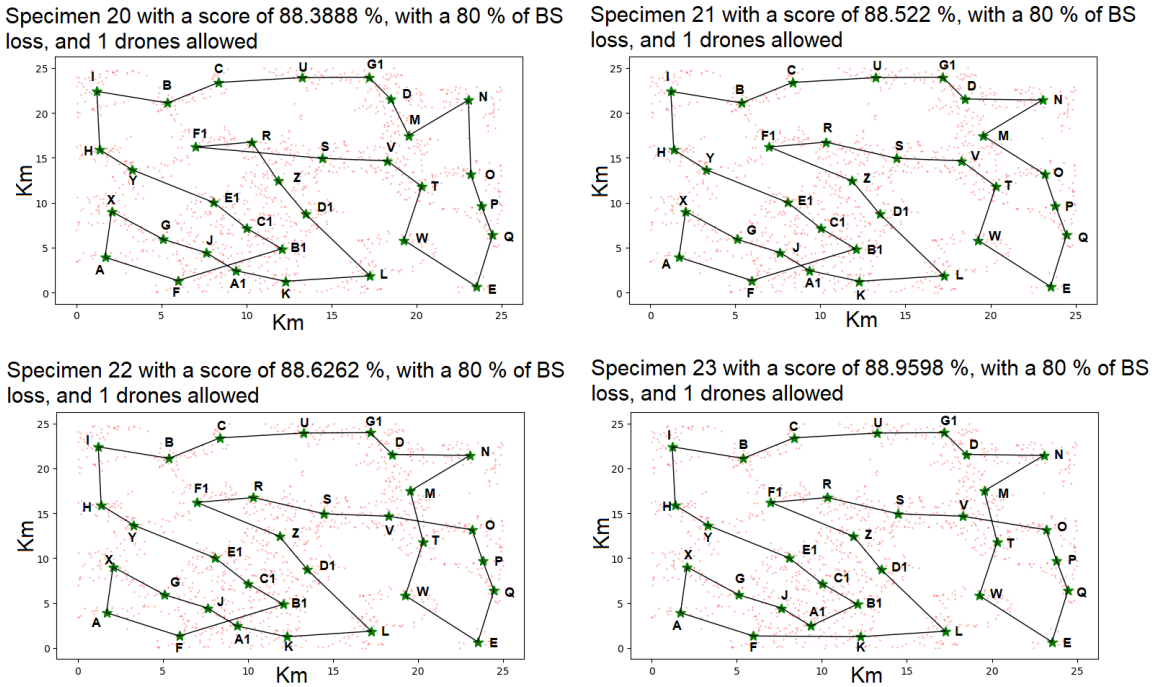
Finally, specimen 23 with a score of 88.9598% is selected as the final specimen. The presented flying loop comprising centroids M, T, W, E, Q, P, O, and V provides a better flyable path than if the graph were completely open whilst not sacrificing service coverage. By enabling such loop the angle $\angle NMO$ is replaced by a path segment that allows a better loop. Thus, the achieved path for the MABS will be $I \rightarrow B \rightarrow C \rightarrow U \rightarrow G1 \rightarrow D \rightarrow N \rightarrow M \rightarrow T \rightarrow W \rightarrow E \rightarrow Q \rightarrow P \rightarrow O \rightarrow V \rightarrow S \rightarrow R \rightarrow F1 \rightarrow Z \rightarrow D1 \rightarrow L \rightarrow K \rightarrow F \rightarrow A \rightarrow$

Figure 4.18: UBPP with 80% BS Loss and 4 MABS Allowed, Part 5



$X \to G \to J \to A1 \to B1 \to C1 \to E1 \to Y \to H \to I.$

Figure 4.19: UBPP with 80% BS Loss and 4 MABS Allowed, Part 6



Specimen 0 in Figure 4.20 provides a scenario where 4 drones are allowed with a 60% of BS loss for a UBNN. Four scrambled paths, which contain the

centroids $ABDEFO$, $NLMGKV$, $HIUST$, and $CQPRJ$. The specimen presents a score of 78.4826%.

Specimen 1 presents two clear flyable paths and another path of scrambled nodes. The paths are $A \to C \to D \to F \to E \to B \to A$ and $N \to K \to M \to L \to N$, while the scrambled path is made with centroids $GHIJUVOPQRST$. While 4 drones are allowed, this specimen evolved to use three for this iteration.

Specimen 2 presents 5 paths, $A \to C \to D \to F \to E \to B \to A$, $N \to K \to M \to L \to N$, $G \to H \to I \to J \to G$, $V \to O \to P \to V$, and $U \to Q \to R \to S \to T \to U$. Since the current best specimen presents more drones than the allowed amount, more generations will be needed to achieve a number of four or lower drones (i.e. paths).

Specimen 3 presents the same paths presented by 2. This happens when two different specimens arrive at the same solution. In this case, the specimens will still be considered for the reproduction population, and the limit counter (i.e., the number of generations that a specimen must stay as the best specimen) of 40 generations will be set back to zero. This will allow any bifurcations from those specimens to keep changing in the evolutionary pool.

Specimens 4, 5, 6, and 7 in Figure 4.21 present a new set of paths, $A \to C \to D \to F \to E \to B \to A$, $H \to I \to K \to M \to L \to N \to G \to H$, $J \to U \to S \to T \to J$, and $V \to O \to P \to Q \to R \to V$. They present the same evolutionary conditions explained for specimens 2 and 3. However, they comply with maximum the amount of drones for the deployment. Here, specimen 7 has a score of 86.07%.

Specimen 8 in Figure 4.22 with a score of 88.722% now requires three drones. It presents a equal paths with specimen 9. The paths,
$A \to C \to D \to F \to E \to B \to A$,
$H \to I \to J \to K \to L \to M \to N \to G \to H$, and

Figure 4.20: UBNN with 60% BS Loss and 4 MABS Allowed, Part 1

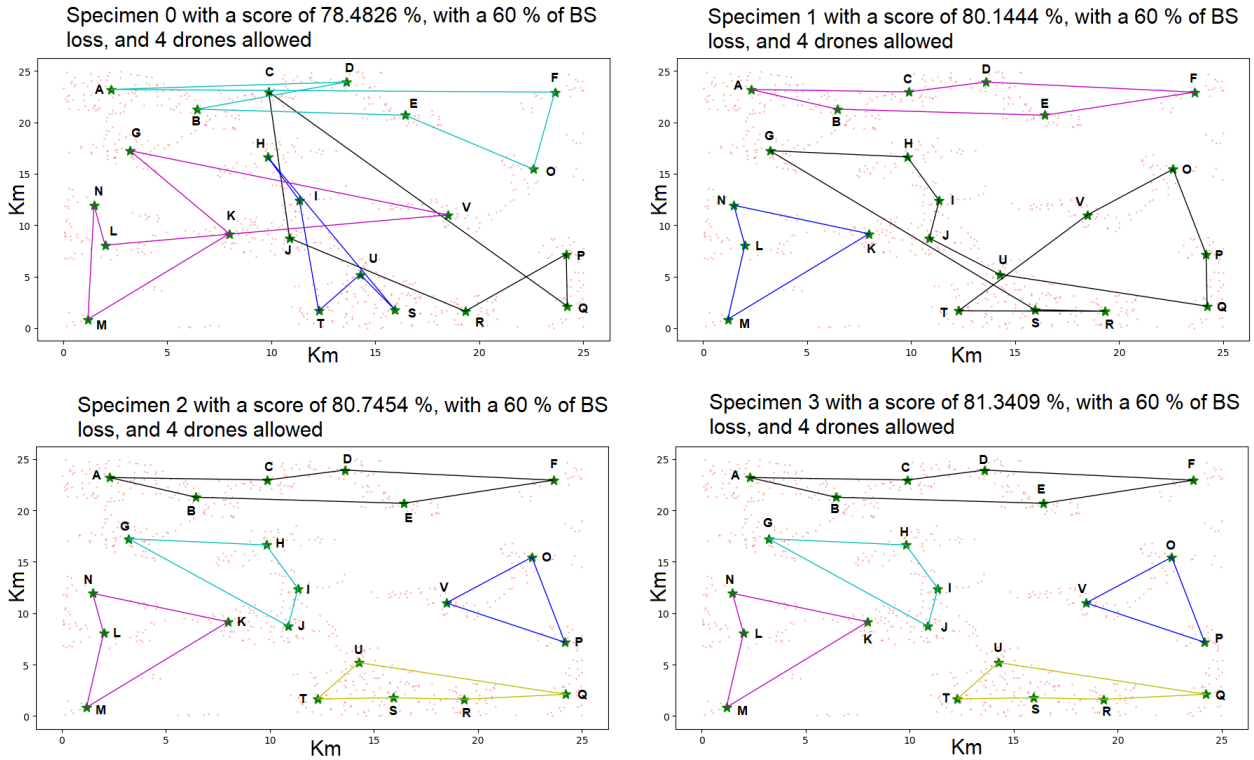
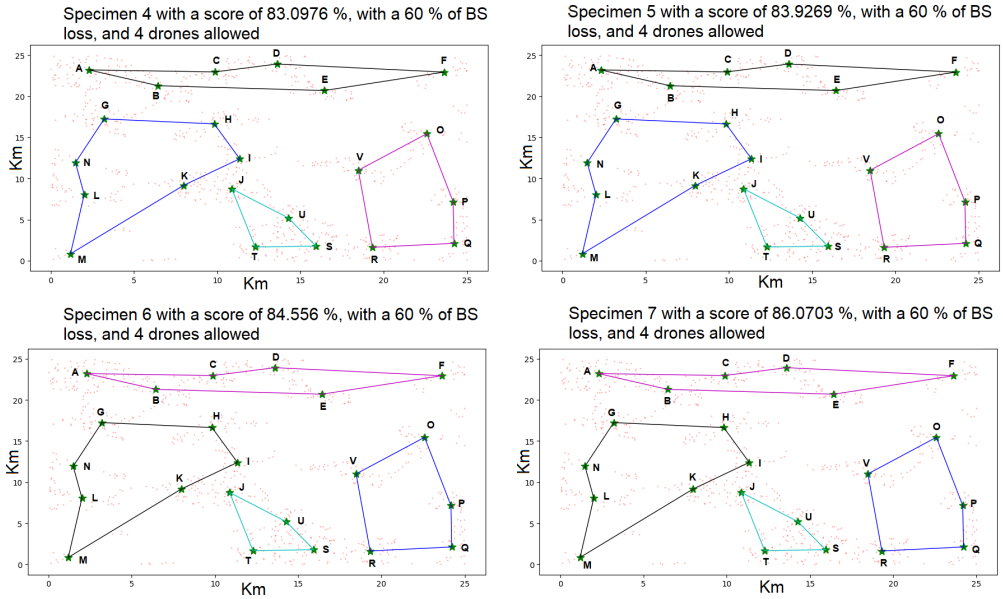


Figure 4.21: UBNN with 60% BS Loss and 4 MABS Allowed, Part 2



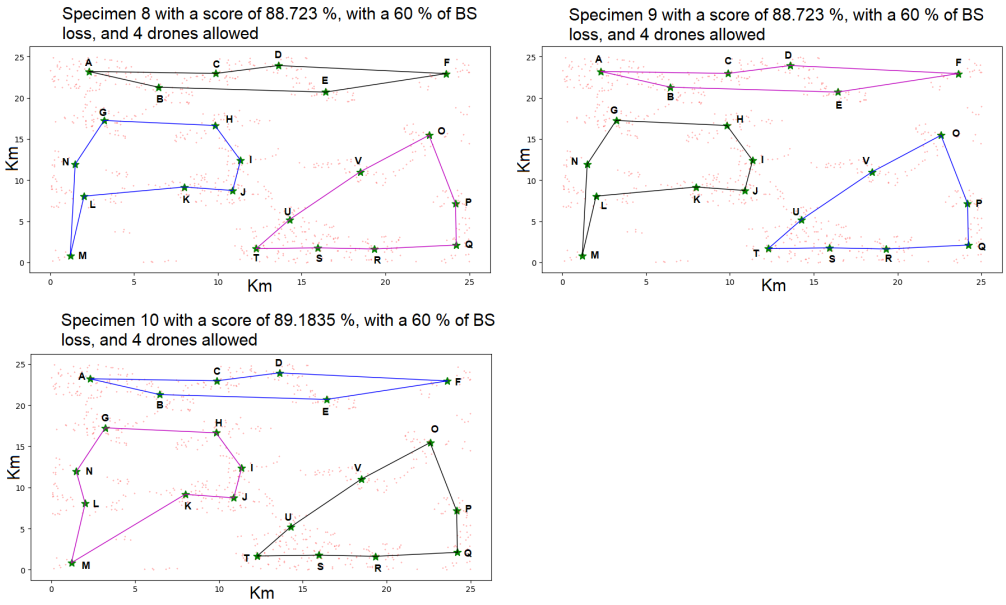
$O \to P \to Q \to R \to S \to T \to U \to O.$

Finally, specimen 10 with a score 89.18% presents a small variation of paths regarding specimen 9. The paths are, $A \to C \to D \to F \to E \to B \to A$,

$H \rightarrow I \rightarrow J \rightarrow K \rightarrow M \rightarrow L \rightarrow N \rightarrow G \rightarrow H$, and

$O \rightarrow P \rightarrow Q \rightarrow R \rightarrow S \rightarrow T \rightarrow U \rightarrow V \rightarrow O$. This specimen is selected as the best and final specimen by using 3 out of the four available drones.

Figure 4.22: UBNN with 60% BS Loss and 4 MABS Allowed, Part 3



Since the specimen number is not related to the generation, it was created, nor how many generations had to pass to achieve its place as the best specimen, we provide some examples to show how many generations a set of a pool of specimens might take to evolve into an acceptable solution.

In Figure 4.23 we can observe a deployment for UBPP with a 80% of BS loss, and 1 drone allowed. Specimen 0 with a score of 77.59% was born in the first generation and replaced by a better specimen in generation 5.

In Figure 4.24 we can observe specimen 29 as the final selected solution for the mentioned scenario. This specimen with a score of 89.57% was born in generation 126 and was selected as the final best specimen in generation 172. This means that the specimen while born in 126 it kept mutating while another specimen was deemed as the current best specimen. Then, in generation 132 after some mutations, it was selected as the best specimen, and 40 generations afterward (172) selected as the final best specimen for the given scenario.

Figure 4.23: UBPP with 80% BS Loss and 1 MABS Allowed, Part 1

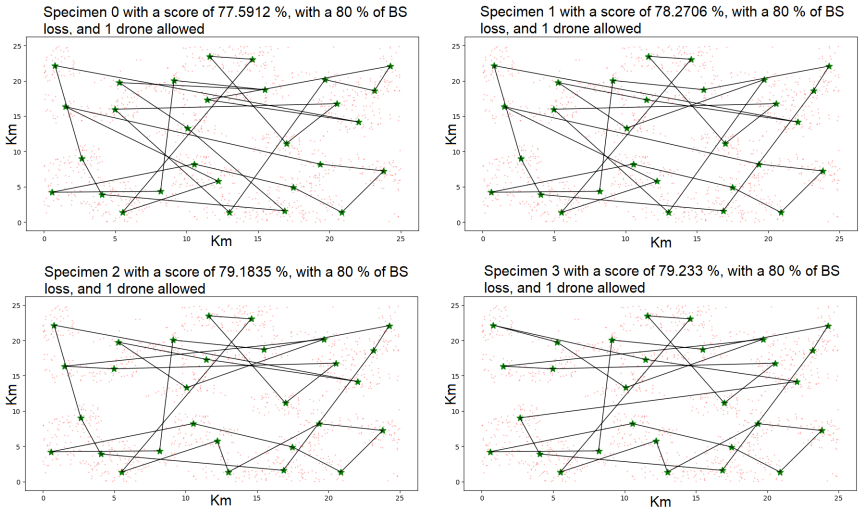
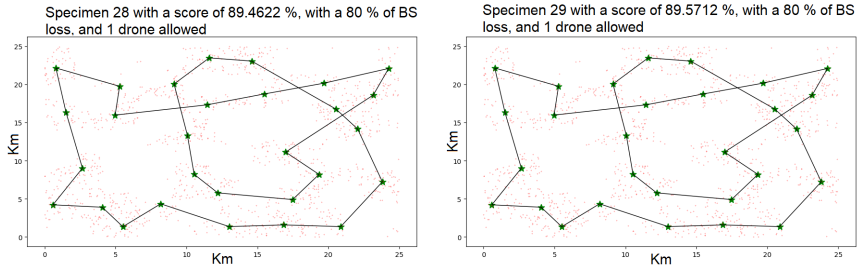
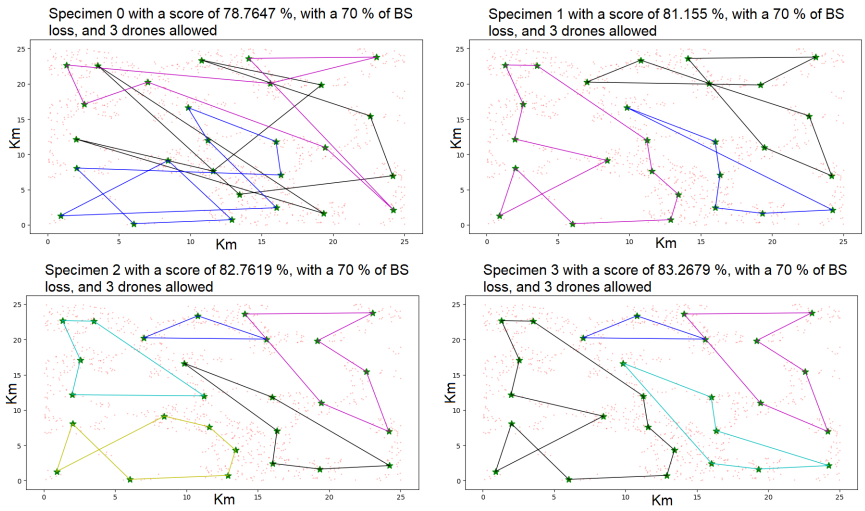


Figure 4.24: UBPP with 80% BS Loss and 1 MABS Allowed, Part 2



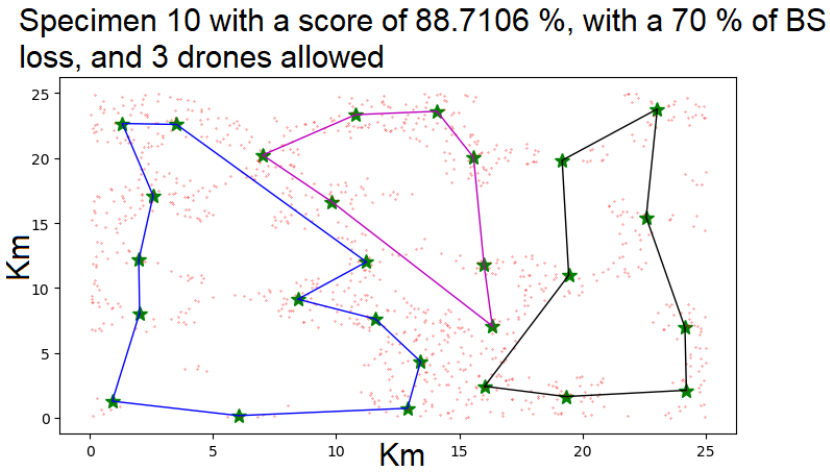
In Figure 4.25 we can observe a deployment for UBNN with a 70% of BS loss, and 3 drones allowed. Specimen 0 with a score of 78.48% was born in the first generation and replaced by a better specimen in generation 4.

Figure 4.25: UBNN with 70% BS Loss and 3 MABS Allowed, Part 1



Specimen 10 in Figure 4.26 with a score of 89.18% was born in generation 41, and selected as the final best specimen in generation 82. This means the specimen had to mutate for one generation until it was selected as the best specimen. This specimen uses the maximum amount of drones at its disposal to provide the utmost adequate service. For this solutions, almost all path segments are perfect flyable paths except for the indentation found on the right side of the left path.

Figure 4.26: UBNN with 70% BS Loss and 3 MABS Allowed, Part 2



In Figure 4.25 we can observe a deployment for UBUU with a 70% of BS loss, and 3 drones allowed. Specimen 0 with a score of 68.23% was born in the first generation and replaced by a better specimen in generation 4.

Specimen 9 in Figure 4.28 with a score of 89.05% was born in generation 14, and selected as the final best specimen in generation 63. This means the specimen had to mutate for 9 generations before being selected as the best specimen. This specimen uses two out of the three available drones. This may be due to a low amount of centroids to visit, and more were not being efficient enough or creating many intersections as seen in specimens 0 and 3 respectively.

Further analysis concerning the number of generations needed to obtain a final best specimen is provided in section 4.2.4.

Figure 4.27: UBUU with 70% BS Loss and 3 MABS Allowed, Part 1

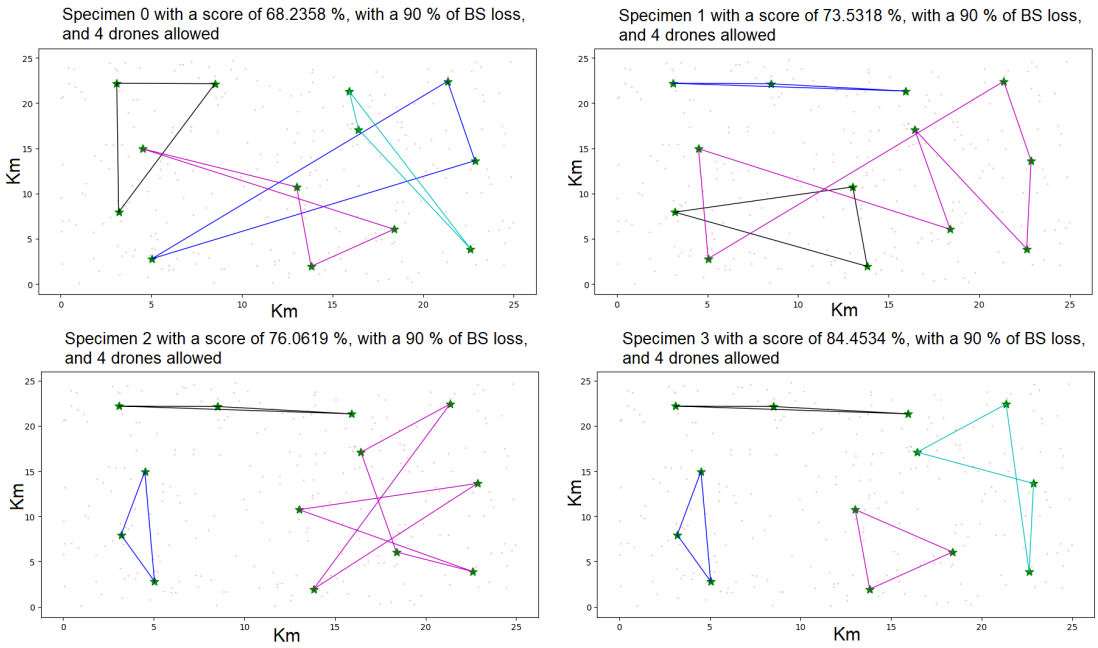
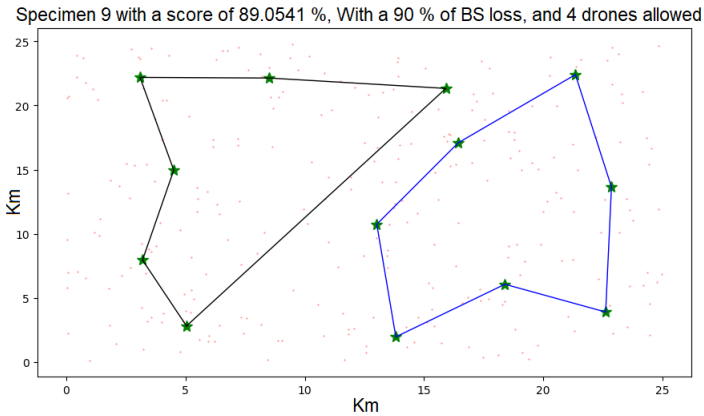


Figure 4.28: UBUU with 70% BS Loss and 3 MABS Allowed, Part 2



4.2.4 Performance Results Comparisons for the proposed MABS Algorithm

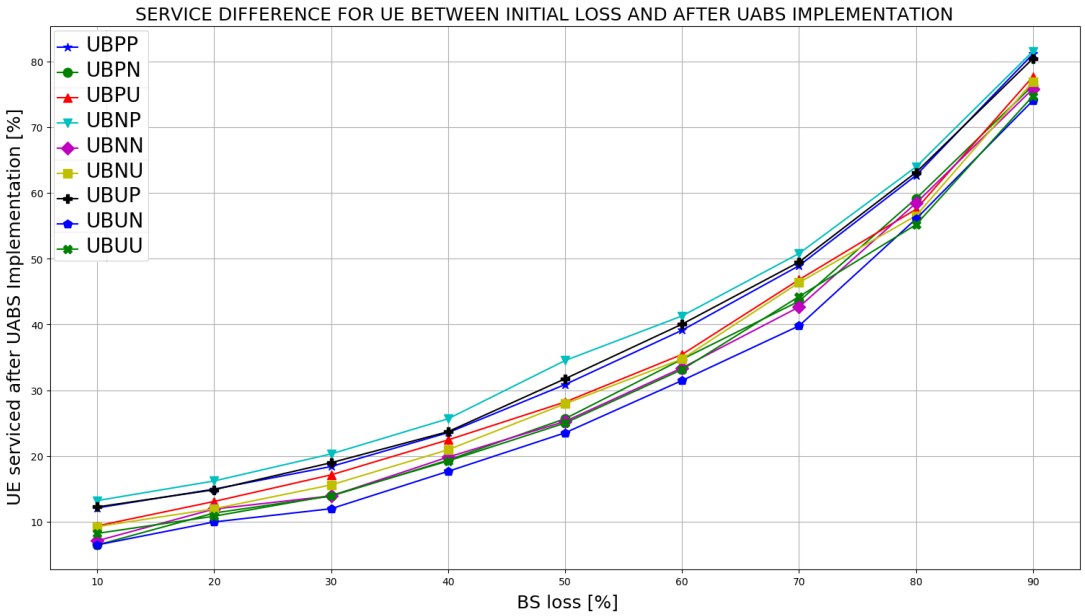
In this section visual representations are provided to analyze the influence of BS loss on how the algorithm provides a solution. To obtain comparison results, 100 simulation results were averaged.

4.2.4.1 Successful Service Achieved by the proposed MABS Algorithm

In this section, we provide an overview of the successfully achieved service by the different distribution combinations described in Table 4.1.

In Figure 4.29 we can observe the percentile of UE that is provided with service after a MABS solution deployment for all considered distributions. All plots show exponential growth; this is because the amount of UE that lost service after BS loss behaves similarly, but with an opposite concave curve. These results can be compared to the initial considerations presented in Figure 4.9 for a better understanding of the increase of service provided.

Figure 4.29: Service Increase with MABS Implementation

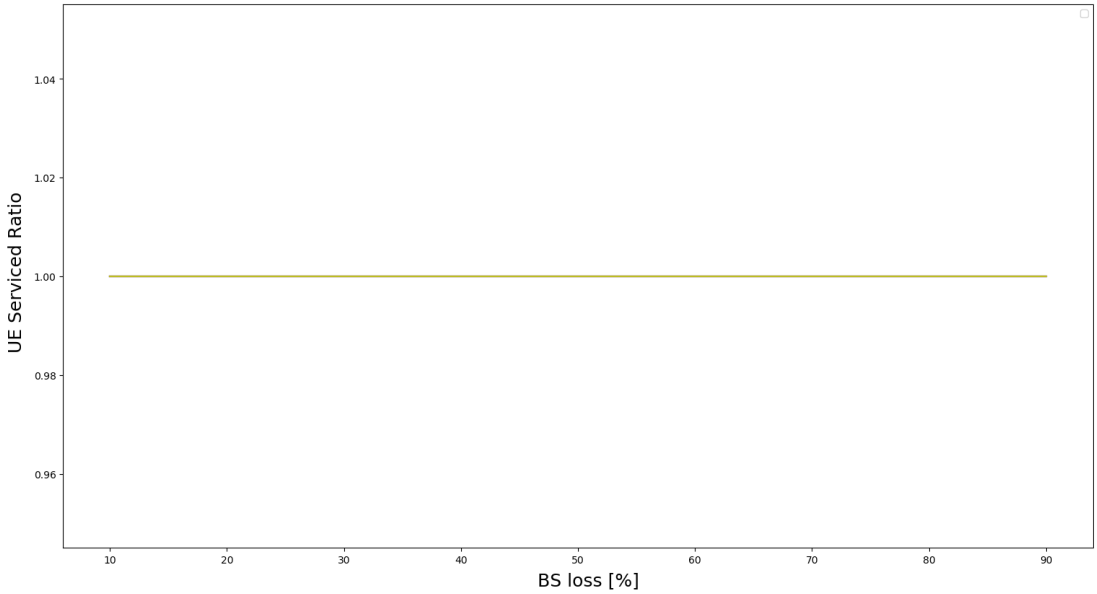


An average ratio of UE that was provided temporal service is shown in Figure 4.30. The single line plotted provides information for all distributions. This is because of how the fitness scoring module was created. By providing a 50% of the total score to service coverage for all specimens, the specimens that didn't provide enough coverage wouldn't be placed between the priority list for further reproduction. Thus, by selection, only solutions that achieved total

service coverage were considered.

In Figure 4.30 we can see a conglomerate of the ratio of all UE with service. A complete coverage is reached as a final result, provided that a UE will at least have a temporal connection regardless of time.

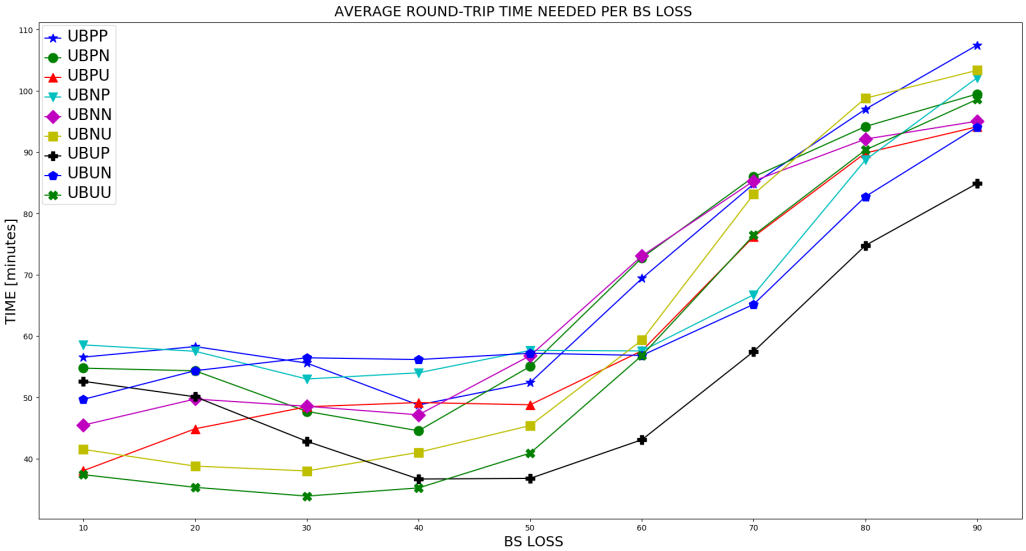
Figure 4.30: Average Ratio of UE with Service After MABS Implementation



In Figure 4.31 we can observe the average amount of time that a MABS takes to visit all centroids before looping through it again. For example, for a path $A \rightarrow B \rightarrow C \rightarrow A$, the time calculated is for the whole path divided by the 50 Km/h assumed for these scenarios. For multiple path solutions, the path with greater distance was considered for the average calculation for that given specimen. Thus, the average is a representation of maximum path lengths within the different solutions.

All solutions appear to behave constantly until there is a 50% BS loss. After that, the amount of time required starts to rise to above an hour. This may be because of the density of UE start to considerably rise above this point in addition to their location spread, thus creating the need to visit more centroids. This is considering that below 50% a MABS may cover unserviced UE in its a path without having a centroid assigned to it.

Figure 4.31: Average Round-Trip Time Cycle for Every Distribution

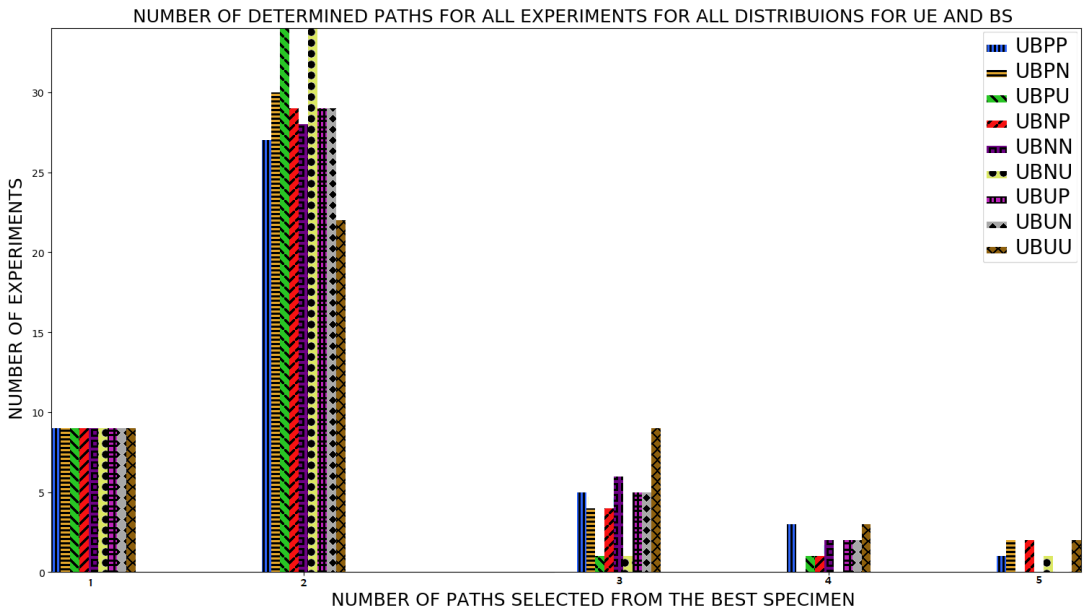


In Figure 4.32 the number of MABSs needed for every distribution is shown. A clear preference for two drones over other solutions is clear. This is in relation to the explanation provided for Figure 4.31. Given that below 50% of BS, the number of time required is constant. Therefore the number of MABSs required doesn't necessarily need to increase to achieve the best solution. For example, in a scenario with 9 centroids, 5 MABS may be too many for an optimum deployment. However, this is not reciprocal, for example in a scenario with many centroids the best solution might be to use fewer drones than available to avoid flyable path intersections or to achieve an overall simpler flyable path to provide a better solution.

With one MABS, the number is constant, since the algorithm will only use one MABS only if forced to, considering that by using less MABSs may recur to a waste of resources. Therefore, a total number of 9 drones is presented (i.e., one for every distribution combination presented in 4.1).

For the rest of MABS (i.e., 3,4,5), the number adapts to the deployed solutions. However, it is clear that not many solutions require 5 MABS to provide the desired service regardless of the distribution combination.

Figure 4.32: Number of Determined PAths for All Experiments for All Distributions



4.2.4.2 Specimen Selection Analysis for proposed MABS Algorithm

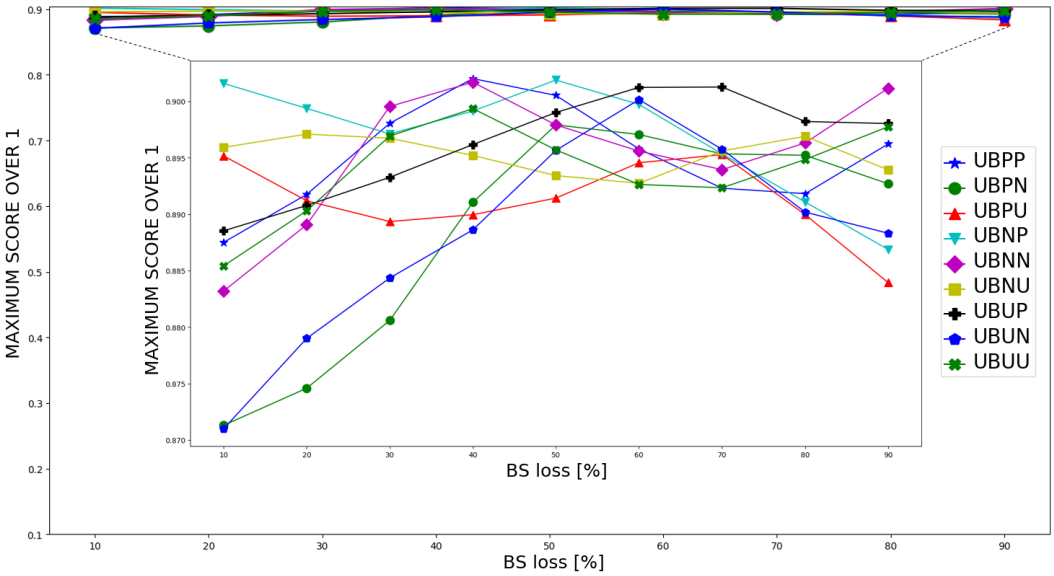
In this section, we analyze how the algorithm achieves the goal of providing optimized paths for different scenarios.

In Figure 4.33 we show an average score achieved by a specimen given the distribution and the loss percentage. As it can be observed, final scores are above 87% regardless of the distribution. As seen, the specimens would achieve a high score but usually stay beneath 95%. This is because for a perfect specimen to exist, a minimum distance, minimum intersections, and angles equal to 180 degrees should be achieved in comparison to other specimens in the evolutionary pool. These speculations are impossible since a specimen would sacrifice not being the one with the shortest distance and maybe have a few interceptions to achieve full coverage. As for the angles, the possible solution for it would be a path where all the points are co-linear. Such a solution doesn't exist in a real-life deployment, thus limiting the overall score to never reach a 100%.

In addition, the average levels presented as final maximum scores achieve similar results at different BS loses for every distribution. This concludes that no

distribution is prone to receive a better score given the current conditions for the experiment set.

Figure 4.33: Maximum Score Per BS Loss Percentage Experiments



The number of candidates needed is a reference to how many best specimens tries to place themselves as the final best solution before getting replaced by another contender or by the unique final best specimen. In Figure 4.34 a maximum number of candidates per BS loss for every distribution can be observed. This provides the reference for how many candidates are needed until no new contender is chosen after 40 generations.

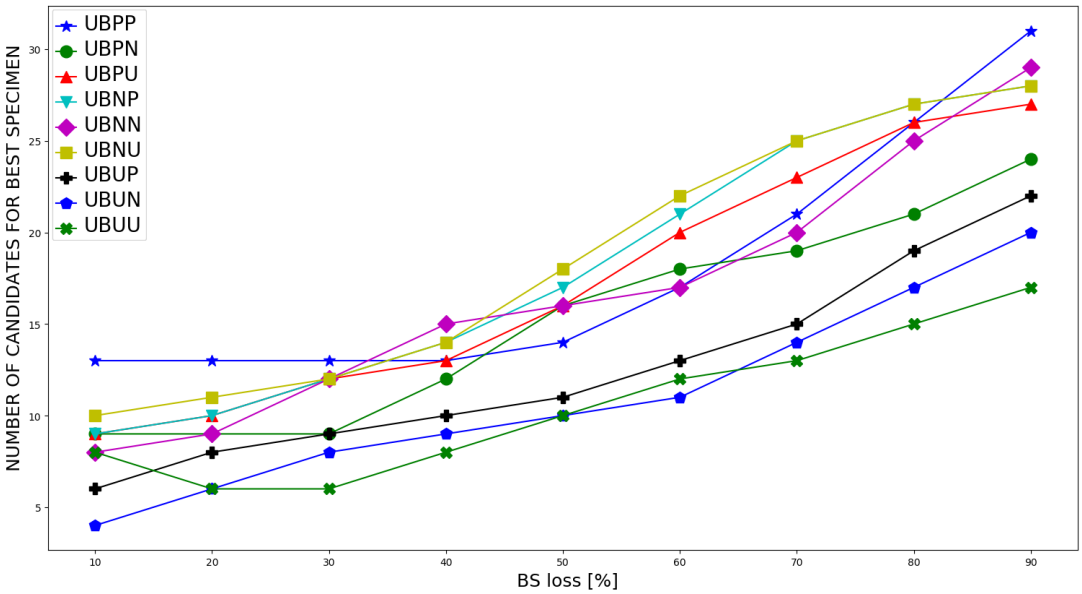
Here it is clear how UE sets created with a Uniform distribution tend to stay lower than other distribution combinations. While they may be an average value, a single experiment has a uniform probability of behaving as other distributions.

On the other hand, UBPP and UBNU keep most of the highest number of candidates required to find the best specimen overall BS loss percentages.

Overall, all curves show an upwards growth on average, which concludes a direct relation between BS loss and a larger number of candidates needed to find the best specimen.

In addition, a high level of mutability means that for a solution, there may

Figure 4.34: Number of Candidates Needed to Find the Best Specimen for Experiments

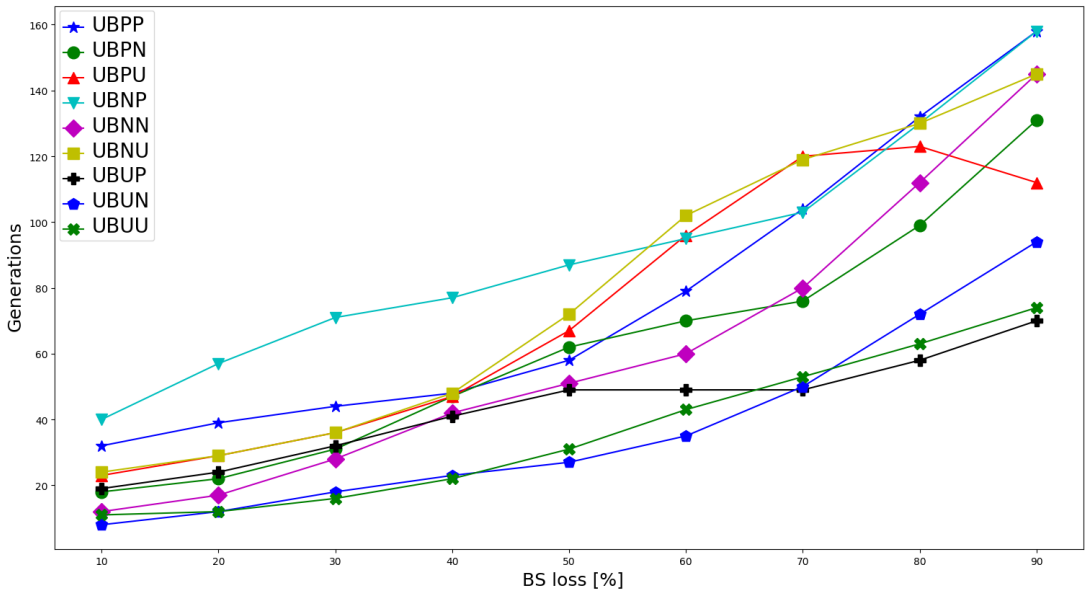


be many new contenders for the best specimen, yet each one can only hold it's the position for a few numbers of generations. Also, a high level of generations required provides a notion of high improvement for a new contender. This is because when a new contender becomes the current best specimen, it will be able to hold its' position for up to 40 generations before being position as a final best specimen. If a new contender takes its place in a number of generations close to 40, it means that the new contender had to present a better solution to the problem. These two introduced notions allow a balance between mutability and quality of mutation to ensure that the best optimum solution is found.

In Figure 4.35 the maximum number of generations to achieve the best specimen is shown. Because the best candidate is only achieved after 40 generations of no best candidate found, the numbers shown in this picture will always be greater than the ones shown in Figure 4.34.

The multiple presented graphs grow with an upper concave on average for all distribution combinations. However, it is noticeable how UE set based on Uniform distribution tend to stay below other distribution combinations.

Figure 4.35: Maximum Number of Generations Needed to Find a Best Specimen



The proposed models would outperform models described in sections 2.6 and 2.6.1 given the requirement of the describes scenario. While other models may offer an increment in bandwidth, they would require additional logistical effort or more machinery to achieve similar results. Thus, the proposed model would provide a better solution with a lower capital and operational expenditure.

Chapter 5

Conclusions

In this thesis, we studied current Disaster Recovery Networks (DRN) usage in post-disaster emergency situations and proposed a solution utilizing Mobile Aerial Base Stations (MABS) paths to enable periodic communication.

First, a primary stage emergency scenario is described as an example of a situation where User Equipments (UE) would require periodic but not necessarily continuous communications. We provided a review of DRN technologies that would be able to assist with such necessities and analyze the benefits and disadvantages that they portray. We surveyed work conducted in DRN organization and deployment, Land Mobile Radio (LMR) deployments, Satellite Communications (SatCom) emergency applications, Cell on Wheels (COW) allocation methods, Device to Device (D2D) communications for public safety, and Unmanned Aerial Vehicles (UAV) implementations for emergency situations. We summarize he analysis by determining the impact of the advantages and disadvantages that would have in post-emergency situations.

We summarized the analysis by determining the advantages and disadvantages that each technology would infer into a post-emergency situation.

Afterward, a Mobile Aerial Base Station (MABS) determining path was proposed as a solution for geographically spread UE with a need for periodic communication. To begin, an Affinity Propagation Clustering algorithm was applied to determine centroids based on a set of clustered CUEs that are left

with no service after a simulated escalatory BS loss. These centroids would then become the visiting nodes (vertices in graph theory) for the MABS solution. With defined centroids, a Multiple Traveler Salesman Problem (MTSP) was created where each centroid must be considered once as a visiting point in the path of a MABS. To determine the best path, a combination of graph theory analysis and a Genetic Algorithm (GA) was used. A solution has been represented by a specimen with multiple chromosomes. Each chromosome represented a different MABS path and was constructed by non-repetitive centroids within its set of chromosomes. The GA implementation considered in-chromosome and cross-chromosome mutation functions that would avoid local solutions and provide the best approach to a global solution for the MTSP problem. In addition, to determine the best-fitted specimen for the proposed problem, fitness functions were implemented. These fitness functions considered Offered Service Coverage, Intra-Path Flight Angles, distance traveled by MABS, and inter-MABS path intersections to determine which candidates may keep reproducing within the algorithm.

To provide a more realistic approach, we implemented the algorithm so that the UE and BS densities would be determined by Poisson, Normal, and Uniform distributions. For every combination, a limited reproduction pool was implemented to ensure that only the best specimens may keep reproducing for every new reproduction cycle. After continuous mutation and reproduction selection cycles, the best specimen would be selected based on a ponderated score of the fitness functions. In addition, each implementation was expanded to allow one to five MABS. This yielded heterogeneous solutions for every Mobile User and BS distribution combination, and the number of MABS available.

Furthermore, the presented data provided a proof for an accurate solution given a described scenario. The proposed computational models and procedures yielded an efficient level of execution time and computation power, considering

the uncertainty and complexity of some scenarios with a percentage of BS loss, spread UE, and available MABS.

Therefore, by employing flyable paths through the proposed algorithm, it was possible to provide periodic communication for geographically spread UE. This would enable enough communication interaction with public safety personnel over an Internet Relay Chat (IRC). While this solution may not be an adequate solution for a data-heavy consumption scenario, it achieves the goal of providing a solution for a more realistic scenario where, during a primary stage of an emergency scenario, resources are scarce and the amount of available help is uncertain.

5.1 Future Work

While the employed algorithm was successful, much additional consideration can be included to enhance the applicability of the proposed solution. For example, external flying conditions (e.g., wind speed, rain), ABS usability and battery life, required communication frequency of UE (i.e., complete technological compatibility), heterogeneity or homogeneity of MABS, differentiated flyable speeds and altitudes for different MABS, and Medium Wireless Access Technologies are some examples.

A meta-heuristic algorithm cannot achieve a result that establishes a solution given the described considerations, yet it provides a background for future solutions to use enhanced methods to consider many more variables. For example, a reinforcement algorithm may be employed to determine paths by not sacrificing time-execution performance and adding more constraints to the proposed problem.

Bibliography

[1] T. Pecorella, L. S. Ronga, F. Chiti, S. Jayousi, and L. Franck. Emergency satellite communications: research and standardization activities. *IEEE Communications Magazine*, 53(5):170–177, May 2015.

[2] E. Del Re, S. Jayousi, S. Morosi, L. S. Ronga, M. De Sanctis, E. Cianca, M. Ruggieri, E. Falletti, A. Iera, G. Araniti, and C. Sacchi. Salice project: Satellite-assisted localization and communication systems for emergency services. *IEEE Aerospace and Electronic Systems Magazine*, 28(9):4–15, Sep. 2013.

[3] Rudolf Giffinger, Christian Fertner, Hans Kramar, Evert Meijers, et al. City-ranking of european medium-sized cities. *Cent. Reg. Sci. Vienna UT*, pages 1–12, 2007.

[4] Pierre Marie Barbara Roussel. Exploring emergency communication experiences in response to the 2010 haiti earthquake: Lessons learned from international donors and local government agencies. 2016.

[5] Goldstein H. Engineers race to restore communications after haiti quake. *IEEE Spectrum, http://spectrum. ieee. org/tech-talk/telecom/internet/engineers-race-to-restorecommunications-after-haiti-quake (20.10. 2014)*, 2010.

[6] Anne Wein, Sally Potter, Sarb Johal, Emma H Doyle, and Julia Becker. Communicating with the public during an earthquake sequence: Improving communication of geoscience by coordinating roles. *Seismological Research Letters*, 87, 11 2015.

[7] Mizobata Y Yamamura H, Kaneda K. Communication problems after the great east japan earthquake of 2011. *Disaster medicine and public health preparedness*, 8, 2014.

[8] Karen Denice Allison Blakely Matt Smith, Eliott C. McLaughlin. Tight airspace, poor communications hinder aid effort, 2010.

[9] C. B. Nelson, B. D. Steckler, and J. A. Stamberger. The evolution of hastily formed networks for disaster response: Technologies, case studies, and future trends. In *2011 IEEE Global Humanitarian Technology Conference*, pages 467–475, Oct 2011.

[10] Z. Kaleem, M. H. Rehmani, M. A. Imran, M. Z. Shakir, A. Jamalipour, and E. Ahmed. Ieee access special section editorial: Mission critical public-safety communications: Architectures, enabling technologies, and future applications. *IEEE Access*, 6:79258–79262, 2018.

[11] A. Kumbhar, F. Koohifar, İ. Güvenç, and B. Mueller. A survey on legacy and emerging technologies for public safety communications. *IEEE Communications Surveys Tutorials*, 19(1):97–124, Firstquarter 2017.

[12] S. Xu, Y. Qian, and R. Qingyang Hu. Reliable and resilient access network design for advanced metering infrastructures in smart grid. *IET Smart Grid*, 1(1):24–30, 2018.

[13] S. Xu, Y. Qian, and R. Q. Hu. On reliability of smart grid neighborhood area networks. *IEEE Access*, 3:2352–2365, 2015.

[14] S. Xu, Y. Qian, and R. Q. Hu. A study on communication network reliability for advanced metering infrastructure in smart grid. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*, pages 127–132, Nov 2017.

[15] S. Xu and Y. Qian. Quantitative study of reliable communication infrastructure in smart grid nan. In *2015 11th International Conference on the Design of Reliable Communication Networks (DRCN)*, pages 93–94, March 2015.

[16] Qian Hu and Naim Kapucu. Information communication technology utilization for effective emergency management networks. *Public Management Review*, 18(3):323–348, 2016.

[17] Naim Kapucu and Vener Garayev. Designing, managing, and sustaining functionally collaborative emergency management networks. *The American Review of Public Administration*, 43(3):312–330, 2013.

[18] Pauliina Palttala, Camillo Boano, Ragnhild Lund, and Marita Vos. Communication gaps in disaster management: Perceptions by experts from governmental and non-governmental organizations. *Journal of Contingencies and Crisis Management*, 20(1):2–12, 2012.

[19] Abraham MartíN-Campillo, Jon Crowcroft, Eiko Yoneki, and Ramon Martí. Evaluating opportunistic networks in disaster scenarios. *Journal of Network and computer applications*, 36(2):870–880, 2013.

[20] S. Ghafoor, P. D. Sutton, C. J. Sreenan, and K. N. Brown. Cognitive radio for disaster response networks: survey, potential, and challenges. *IEEE Wireless Communications*, 21(5):70–80, October 2014.

[21] DG Reina, Jose Maria León-Coca, SL Toral, Eleana Asimakopoulou, Federico Barrero, Peter Norrington, and Nik Bessis. Multi-objective performance optimization of a probabilistic similarity/dissimilarity-based broadcasting scheme for mobile ad hoc networks in disaster response scenarios. *Soft Computing*, 18(9):1745–1756, 2014.

[22] Lili Yang, Shuang-Hua Yang, and Linda Plotnick. How the internet of things technology enhances emergency response operations. *Technological Forecasting and Social Change*, 80(9):1854–1867, 2013.

[23] Seppo Horsmanheimo, Niwas Maskey, Lotta Tuomimäki, and Kari Mäki. Interoperability of electricity distribution and communication networks in large-scale outage situations. In *2015 IEEE International Telecommunications Energy Conference (INTELEC)*, pages 1–6. IEEE, 2015.

[24] C. B. Nelson, B. D. Steckler, and J. A. Stamberger. The evolution of hastily formed networks for disaster response: Technologies, case studies, and future trends. In *2011 IEEE Global Humanitarian Technology Conference*, pages 467–475, Oct 2011.

[25] Xuesong Guo and Naim Kapucu. Examining collaborative disaster response in china: network perspectives. *Natural Hazards*, 79(3):1773–1789, 2015.

[26] Michel Daoud Yacoub. *Foundations of mobile radio engineering*. Routledge, 2019.

[27] Eran Greenberg and Shachar Greenberg. Simulations and measurements of the radio channel propagation parameters in urban and industrial

environments for emergency localization. In *2011 IEEE International Conference on Microwaves, Communications, Antennas and Electronic Systems (COMCAS 2011)*, pages 1–7. IEEE, 2011.

[28] Tanim M Taher, Roger B Bacchus, Kenneth J Zdunek, and Dennis A Roberson. Dynamic spectrum access opportunities for public safety in land mobile radio bands. In *2011 6th International ICST Conference on Cognitive Radio Oriented Wireless Networks and Communications (CROWNCOM)*, pages 355–359. IEEE, 2011.

[29] Abhaykumar Kumbhar and Ismail Güvenç. A comparative study of land mobile radio and lte-based public safety communications. In *SoutheastCon 2015*, pages 1–8. IEEE, 2015.

[30] M. Berioli, N. Courville, and M. Werner. Emergency communications over satellite: the wisecom approach. In *2007 16th IST Mobile and Wireless Communications Summit*, pages 1–5, July 2007.

[31] Matteo Berioli, Javier Mulero Chaves, Nicolas Courville, Philippe Boutry, Jean-Louis Fondere, Harald Skinnemoen, Hillar Tork, Markus Werner, and Michael Weinlich. Wisecom: A rapidly deployable satellite backhauling system for emergency situations. *International Journal of Satellite Communications and Networking*, 29(5):419–440, 2011.

[32] Matteo Berioli, Nicolas Courville, and Markus Werner. Integrating satellite and terrestrial technologies for emergency communications: the wisecom project. In *QShine 2007 Workshop: Satellite/Terrestrial Interworking*, page 2. ACM, 2007.

[33] M. Berioli, A. Molinaro, S. Morosi, and S. Scalise. Aerospace communications for emergency applications. *Proceedings of the IEEE*, 99(11):1922–1938, Nov 2011.

[34] E. H. Fazli, M. Werner, N. Courville, M. Berioli, and V. Boussemart. Integrated gsm/wifi backhauling over satellite: Flexible solution for emergency communications. In *VTC Spring 2008 - IEEE Vehicular Technology Conference*, pages 2962–2966, May 2008.

[35] Francesco Chiti, Romano Fantacci, Leonardo Maccari, Dania Marabissi, and Daniele Tarchi. A broadband wireless communications system for emergency management. *IEEE Wireless Communications*, 15(3):8–14, 2008.

[36] and, , , and and. Emergency communication system by heterogeneous wireless networking. In *2010 IEEE International Conference on Wireless Communications, Networking and Information Security*, pages 488–492, June 2010.

[37] Y. Lee and B. Ku and. A satellite core network system for emergency management and disaster recovery. In *2010 International Conference on Information and Communication Technology Convergence (ICTC)*, pages 549–552, Nov 2010.

[38] J. Košmerl and A. Vilhar. Base stations placement optimization in wireless networks for emergency communications. In *2014 IEEE International Conference on Communications Workshops (ICC)*, pages 200–205, June 2014.

[39] S. Montori, F. Cacciamani, R. V. Gatti, R. Sorrentino, G. Arista, C. Tienda, J. A. Encinar, and G. Toso. A transportable reflectarray antenna for satellite ku-band emergency communications. *IEEE Transactions on Antennas and Propagation*, 63(4):1393–1407, April 2015.

[40] J. Huang and Y. Lien. Challenges of emergency communication network for disaster response. In *2012 IEEE International Conference on Communication Systems (ICCS)*, pages 528–532, Nov 2012.

[41] Supriyo Ghosh and Pradeep Varakantham. Strategic planning for setting up base stations in emergency medical systems. In *Twenty-Sixth International Conference on Automated Planning and Scheduling*, 2016.

[42] S. Bhattarai, S. Wei, S. Rook, W. Yu, D. Griffith, and N. Golmie. Optimizing the location deployment of dynamic mobile base stations. In *2015 International Conference on Computing, Networking and Communications (ICNC)*, pages 579–583, Feb 2015.

[43] C. Wechtaisong, T. Sutthitep, and C. Prommak. Multi-objective planning and optimization for base station placement in wimax network. In *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, pages 1–4, May 2014.

[44] D. Abusch-Magder, P. Bosch, T. E. Klein, P. A. Polakos, L. G. Samuel, and H. Viswanathan. 911-now: A network on wheels for emergency response and disaster recovery operations. *Bell Labs Technical Journal*, 11(4):113–133, Winter 2007.

[45] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi. Device-to-device communications for national security and public safety. *IEEE Access*, 2:1510–1520, 2014.

[46] J. Liu, N. Kato, J. Ma, and N. Kadowaki. Device-to-device communication in lte-advanced networks: A survey. *IEEE Communications Surveys Tutorials*, 17(4):1923–1940, Fourthquarter 2015.

[47] J. Huang, D. Fang, F. Ye, R. Q. Hu, and Y. Qian. A relay selection scheme to prolong connection time for public safety communications. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5, June 2018.

[48] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi. Device-to-device communications for national security and public safety. *IEEE Access*, 2:1510–1520, 2014.

[49] S. Gyawali, S. Xu, F. Ye, R. Q. Hu, and Y. Qian. A d2d based clustering scheme for public safety communications. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5, June 2018.

[50] K. Ali, H. X. Nguyen, P. Shah, Q. Vien, and N. Bhuvanasundaram. Architecture for public safety network using d2d communication. In *2016 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pages 206–211, April 2016.

[51] M. Usman, A. A. Gebremariam, U. Raza, and F. Granelli. A software-defined device-to-device communication architecture for public safety applications in 5g networks. *IEEE Access*, 3:1649–1654, 2015.

[52] H. Nishiyama, M. Ito, and N. Kato. Relay-by-smartphone: realizing multihop device-to-device communications. *IEEE Communications Magazine*, 52(4):56–65, April 2014.

[53] S. Chandrasekharan, K. Gomez, A. Al-Hourani, S. Kandeepan, T. Rasheed, L. Goratti, L. Reynaud, D. Grace, I. Bucaille, T. Wirth, and S. Allsopp. Designing and implementing future aerial communication networks. *IEEE Communications Magazine*, 54(5):26–34, May 2016.

[54] Y. Zeng, R. Zhang, and T. J. Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, May 2016.

[55] S. Chandrasekharan, S. Kandeepan, R. J. Evans, A. Munari, R. Hermenier, M. Marchitti, and K. Gomez. Clustering approach for aerial base-station

access with terrestrial cooperation. In *2013 IEEE Globecom Workshops (GC Wkshps)*, pages 1397–1402, Dec 2013.

[56] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni. Drone-aided communication as a key enabler for 5g and resilient public safety networks. *IEEE Communications Magazine*, 56(1):36–42, Jan 2018.

[57] Y. Zeng, R. Zhang, and T. J. Lim. Wireless communications with unmanned aerial vehicles: opportunities and challenges. *IEEE Communications Magazine*, 54(5):36–42, May 2016.

[58] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah. Efficient deployment of multiple unmanned aerial vehicles for optimal wireless coverage. *IEEE Communications Letters*, 20(8):1647–1650, Aug 2016.

[59] M. M. Azari, F. Rosas, K. Chen, and S. Pollin. Joint sum-rate and power gain analysis of an aerial base station. In *2016 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, Dec 2016.

[60] 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 *2016 IEEE International Conference on Communications (ICC)*, pages 1–5, May 2016.

[61] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu. 3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage. *IEEE Wireless Communications Letters*, 6(4):434–437, Aug 2017.

[62] J. Sae, S. F. Yunas, and J. Lempiainen. Coverage aspects of temporary lap network. In *2016 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS)*, pages 1–4, Jan 2016.

[63] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski. Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85, Jan 2017.

[64] M. Marinelli, L. Caggiani, M. Ottomanelli, and M. Dell'Orco. En route truck–drone parcel delivery for optimal vehicle routing strategies. *IET Intelligent Transport Systems*, 12(4):253–261, 2018.

[65] Azade Fotouhi, Ming Ding, and Mahbub Hassan. Dronecells: Improving 5g spectral efficiency using drone-mounted flying base stations. *arXiv preprint arXiv:1707.02041*, 2017.

[66] A. Fotouhi, M. Ding, and M. Hassan. Dynamic base station repositioning to improve performance of drone small cells. In *2016 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6, Dec 2016.

[67] M. Deruyck, A. Marri, S. Mignardi, L. Martens, W. Joseph, and R. Verdone. Performance evaluation of the dynamic trajectory design for an unmanned aerial base station in a single frequency network. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, Oct 2017.

[68] Z. Han, A. L. Swindlehurst, and K. J. R. Liu. Optimization of manet connectivity via smart deployment/movement of unmanned air vehicles. *IEEE Transactions on Vehicular Technology*, 58(7):3533–3546, Sep. 2009.

[69] Z. Han, A. L. Swindlehurst, and K. J. R. Liu. Smart deployment/movement of unmanned air vehicle to improve connectivity in manet. In *IEEE Wireless Communications and Networking Conference, 2006. WCNC 2006.*, volume 1, pages 252–257, April 2006.

[70] Z. Zhaochen and Z. Jing. Relay route planning based on connectivity in airborne ad hoc networks. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 1277–1284, May 2017.

[71] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim. Placement optimization of uav-mounted mobile base stations. *IEEE Communications Letters*, 21(3):604–607, March 2017.

[72] Karina Gomez, Sithamparanathan Kandeepan, Macia Mut Vidal, Vincent Boussemart, Raquel Ramos, Romain Hermenier, Tinku Rasheed, Leonardi Goratti, Laurent Reynaud, David Grace, et al. Aerial base stations with opportunistic links for next generation emergency communications. *IEEE Communications Magazine*, 54(4):31–39, 2016.

[73] Milan Erdelj, Enrico Natalizio, Kaushik R Chowdhury, and Ian F Akyildiz. Help from the sky: Leveraging uavs for disaster management. *IEEE Pervasive Computing*, 16(1):24–32, 2017.

[74] N. Rupasinghe, Y. Yapıcı, İ. Güvenç, and Y. Kakishima. Non-orthogonal multiple access for mmwave drone networks with limited feedback. *IEEE Transactions on Communications*, 67(1):762–777, Jan 2019.

[75] M Sakti Alvissalim, Big Zaman, Z Ahmad Hafizh, M Anwar Ma'sum, Grafika Jati, Wisnu Jatmiko, and Petrus Mursanto. Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area. In *2012 Proceedings of SICE Annual Conference (SICE)*, pages 2256–2261. IEEE, 2012.

[76] Milan Erdelj, Michał Król, and Enrico Natalizio. Wireless sensor networks and multi-uav systems for natural disaster management. *Computer Networks*, 124:72–86, 2017.

[77] N. Rupasinghe, Y. Yapıcı, İ. Güvenç, and Y. Kakishima. Non-orthogonal multiple access for mmwave drone networks with limited feedback. *IEEE Transactions on Communications*, 67(1):762–777, Jan 2019.

[78] A. Kumbhar, S. Singh, and I. Guvenc. Uav assisted public safety communications with lte-advanced hetnets and feicic. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 1–7, Oct 2017.

[79] A. Merwaday, S. Mukherjee, and I. Güvenç. Hetnet capacity with reduced power subframes. In *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1380–1385, April 2014.

[80] A. Kumbhar, İ. Güvenç, S. Singh, and A. Tuncer. Exploiting lte-advanced hetnets and feicic for uav-assisted public safety communications. *IEEE Access*, 6:783–796, 2018.

[81] Theodore S Rappaport. Wireless communications–principles and practice, (the book end). *Microwave Journal*, 45(12):128–129, 2002.

[82] A. Fotouhi, M. Ding, and M. Hassan. Dynamic base station repositioning to improve spectral efficiency of drone small cells. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9, June 2017.

[83] Mohammad Sedighpour, Majid Yousefikhoshbakht, and Narges Mahmoodi Darani. An effective genetic algorithm for solving the multiple traveling salesman problem. *Journal of Optimization in Industrial Engineering*, Volume 4(8):73–79, 2012.

[84] András Király and János Abonyi. *Optimization of Multiple Traveling Salesmen Problem by a Novel Representation Based Genetic Algorithm*, pages 241–269. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[85] Mingji Xu, Sheng Li, and Jian Guo. Optimization of multiple traveling salesman problem based on simulated annealing genetic algorithm. In *MATEC Web of Conferences*, volume 100, page 02025. EDP Sciences, 2017.

[86] J. Li, Q. Sun, M. Zhou, and X. Dai. A new multiple traveling salesman problem and its genetic algorithm-based solution. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*, pages 627–632, Oct 2013.

[87] Ali Shokouhi Rostami, Farahnaz Mohanna, Hengameh Keshavarz, and Ali Asghar Rahmani Hosseinabadi. Solving multiple traveling salesman problem using the gravitational emulation local search algorithm. *Appl. Math*, 9(2):699–709, 2015.

[88] Zhi Ding B.P. Lathi. *Modern Digital ans Analog Communication Systems.* Oxford University Press Inc., 2009.

[89] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[90] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[91] Vitaly I Voloshin. *Introduction to graph theory.* Nova Science Publ., 2009.

[92] Xin-She Yang. *Nature-inspired metaheuristic algorithms.* Luniver press, 2010.

[93] Xin-She Yang. *Engineering optimization: an introduction with metaheuristic applications.* John Wiley & Sons, 2010.

[94] Kaijun Wang, Junying Zhang, Dan Li, Xinna Zhang, and Tao Guo. Adaptive affinity propagation clustering. *arXiv preprint arXiv:0805.1096*, 2008.

[95] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007.

[96] Fred W Glover and Gary A Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.

[97] Fred W Glover and Gary A Kochenberger. *Handbook of metaheuristics*, volume 57. Springer Science & Business Media, 2006.

[98] S. K. Chang. *Data Structures And Algorithms.* Number v. 13 in Series on Software Engineering and Knowledge Engineering. World Scientific, 2003.

[99] Kevin P Murphy. *Machine learning: a probabilistic perspective.* MIT press, 2012.

Appendix A

Glossary of Key Terms

3G	Third Generation Cellular Networks.
3GPP	3rd Generation Partnership Project
4G	Fourth Generation Cellular Networks.
5G	Fifth Generation Cellular Networks.
ADSL	Asymmetric Digital Subscriber.
bps	Bits Per Second
BS	Base Station
Bw	Bandwidth given in Hz
C	Capacity given in bps
CRE	Cell Range Expansion
COW	Cell On Wheels
D2D	Device to Device
Drone	Unmanned autonomous or remotely controlled apparatus.
FeICIC	Further enhanced ICIC

FTTH	Fiber To The Home
GA	Genetic Algorithm
GCB	Geostationary Communications Balloon
HetNet	Heterogeneous Networks
IRC	Internet Relay Chat
ICIC	Inter-Cell Interference Coordination
LAP	Low Altitude Platform
LTE	Long Term Evolution
MBS	Macro Base Station
mmWaves	mili-meter Waves
MTSP	Multiple Traveler Salesman Problem
MOI	Macro-Base Station Of Interest
MABS	Mobile Aerial Base Station
MUABS	Mobile Unmanned Aerial Base Station
PSN	Public Safety Network
QoS	Quality of Service
RBC	Renaming Battery Capacity
REB	Range Expansion Bias
SNR	Signal to Noise Ratio
TSP	Traveler Salesman Problem

UABS Unmanned Aerial Base Station

UAV Unmanned Aerial Vehicle

UE User Equipment

UML Unified Modeling Language

UOI UABS of Interest

VSat Very Small Aperture Terminal

Wi-Fi/WiFi Name used to market 802.11 WLAN technology. Usually also refered as IEEE 802.11 b/g/n/ac.

WiMax Worldwide Interoperability for Microwave Access

Appendix B

Code

Main.py

```
# -*- coding: utf-8 -*-
"""
Main code where all experiments will be carried
@author: Jose Matamoros
"""
from ClusterCentroid import centroids
import matplotlib.pyplot as plt
from scipy.spatial import Voronoi, voronoi_plot_2d
import numpy as np
from AssignedBS import assignedBS
from CapacityCal import capacity
import UAV

class Experiment:
    '''
    Each object will be a different experiment, comprised of:
    -A set of UE point sets
    -A set of working BS
    -An genesis parent from which children will be created
    -Multiple Childs

    The object will carry the results of the expermients within it's local variables.

    '''
    def __init__(self, UEset, BSset, capacity_threshold):
        '''
        Requires a set of MB and UE
        '''
        self.capacity_threshold = capacity_threshold
        self.BSset = BSset
        self.UEset = UEset
        self.capacities = []
        self.unserviced = []
        self.results = []
        # create parent based on given clusters from a set
        # create first generation of childs from parent
```

```
        self.initial_assigned_BS = assignedBS([[x,y] for x,y in zip(UEset.xue,UEset.yue)],[[x,y] for x,y
in zip(BSset.xbs,BSset.ybs)])
        ue_set = list(zip([x for x in UEset.xue], [y for y in UEset.yue]))
        bs_set = list(zip([x for x in BSset.xbs], [y for y in BSset.ybs]))
        self.initial_capacity = capacity(ue_set, bs_set,self.initial_assigned_BS, capacity_threshold)

    def process_loss(self):
                '''
                A process to determine BS loss and all variable sets needed for further
experiments.
                '''
        ue_set = list(zip([x for x in self.UEset.xue], [y for y in self.UEset.yue]))
        loss_set = [i*0.1 for i in range(1,10)]
        tot_l = len(self.BSset.xbs)
        for loss in loss_set:
            BStempX = self.BSset.xbs[:round(tot_l*loss)]
            BStempY = self.BSset.ybs[:round(tot_l*loss)]
            bs_set_temp = list(zip([x for x in BStempX], [y for y in BStempY]))
            temp_assignedBS = assignedBS([[x,y] for x,y in zip(self.UEset.xue,self.UEset.yue)],[[x,y] for
x,y in zip(BStempX,BStempY)])
            self.capacities.append(capacity(ue_set, bs_set_temp, temp_assignedBS,
self.capacity_threshold))
            temp_un=[]
            for user,cap in zip(ue_set, self.capacities[-1]):
                if(cap[1] == 0):
                    temp_un.append(user)
            self.unserviced.append(temp_un)

    def process_uav(self, specimen_population):
        '''
        Executes a set of mutation funcions and thorugh fitness functions retreives the best
candidates.

        Parameters
        ----------
                specimen_population : int
                        The number of best secimens that are selected to keep mutating
between generations.

        Returns
        -------

        '''
        self.unserviced_centroids = []
        for loss in range(0,9):
            unserviced_tmp = self.unserviced[loss][:]
            tmp_unserviced_centroids = centroids([x[0] for x in unserviced_tmp],[x[1] for x in
unserviced_tmp])
```

```
        self.unserviced_centroids.append(tmp_unserviced_centroids)
#         self.uav_set = [UAVset(0,0,i,tmp_unserviced_centroids) for i in range(1,6)]
        self.uav_set = [UAVset(0,0,i,tmp_unserviced_centroids) for i in range(1,2)]
        for n,uav_option in enumerate(self.uav_set):
            best_score = 0
            candidates = []
            fit_counter = 0
            limit = 40
            while(fit_counter<limit):
                fit_counter += 1
                uav_option.mutate()
                uav_option.fit(unserviced_tmp,200000,5, len(tmp_unserviced_centroids),
specimen_population)
                best = uav_option.best_uav
                if(best[0] > best_score):
                    best_score = best[0]
                    candidates.append({'specimen':best[1],'score':best_score})
                    fit_counter = 0
            print('Calculated drone %d out of %d for loss %d out of %d'
%(n+1,len(self.uav_set),loss+1, 9))
            result = {'loss':(9 - loss)*10, 'allowed_drones':(n+1), 'candidates':candidates,
'best_score':best_score, 'best_specimen':candidates[-1]}
            self.results.append(result)

    def showResults(self):
        '''
                    Displays the initial results for the given experiment.

                    Parameters
        ----------

        Returns
        -------

                    '''
        #show initial serviced UE
        fig1 = plt.figure(1)
        plt.hist([x[1] for x in self.initial_capacity])
        plt.xlabel('Service')
        plt.ylabel('Ammount')
        plt.title('Number of UE that have service before Loss')
        fig1.show()

        #show % of serviced UE in relation to BS loss
        prt = []
        for cap in self.capacities:
            [a,b,c] = plt.hist([x[1] for x in cap],2)
            prt.append(a[0]*100/(a[0]+a[1]))
```

```
    fig2 = plt.figure(2)
    plt.plot(([100-x for x in prt[::-1]]),'--r')
    plt.xlabel('BS loss [%]')
    plt.ylabel('UE serviced [%]')
    plt.title('SERVICE TO UE ACCORDING TO NUMBER OF BS')
    plt.grid(True)
    fig2.show()

class UserEquipmentSet:
  '''
  Each object will have a set of points where the UE are, and a set of determinated clusters
according to those points.
  '''
  def __init__(self,density, distribution,area):
    quantity = density * area
    if(distribution == 'p'):
      numue = np.random.poisson(quantity, 1) # lambda = quantity
    elif(distribution == 'n'):
      numue = np.int32(np.abs(np.round(np.random.normal(quantity, 10, 1))))# mu = quantity,
sigma = 10
    elif(distribution == 'u'):
      numue = np.int32(np.abs(np.round(np.random.uniform(0,quantity,1))))

    self.xue = np.random.uniform(0,area,numue)
    self.yue = np.random.uniform(0,area,numue)
    self.clusters = centroids(self.xue, self.yue)

class MacroBaseSet:
  '''
  Each object will have a set of points that establishes where will working base stations be
stablished within an experiment.
  '''
  def __init__(self,density, distribution, area):
    quantity = density * area
    if(distribution == 'p'):
      numbs = np.random.poisson(quantity, 1) # lambda = quantity
    elif(distribution == 'n'):
      numbs = np.int32(np.abs(np.round(np.random.normal(quantity, 10, 1))))# mu = quantity,
sigma = 10
    elif(distribution == 'u'):
      numbs = np.int32(np.abs(np.round(np.random.uniform(0,quantity,1))))

    self.xbs = np.random.uniform(0,area,numbs)
    self.ybs = np.random.uniform(0,area,numbs)

class UAVset:
  '''
```

```
    Each class must be created for each child.
    Each child will have mustiple chromosomes within, which will allow it to mutate.
    '''
    def __init__(self, born, current, drone_limit, centroids):
        self.current = 0
        self.drone_number= drone_limit
        self.uavs = [UAV.Child(current,centroids, drone_limit) for x in range(0,50)]

    def mutate(self):
        '''
        Generates new generations of childs based based on mutators.
        At the end of the process, the UAV object will have more Childs (combinations)

        Parameters
        ----------

        Returns
        -------

        '''
        self.current += 1
        tmp=[]
        for uav in self.uavs:
            for n_uav in uav.mutate(self.current): tmp.append(n_uav)
            #for n_uav in UAV.Child(self.current,uav.mutate(self.current)): self.uavs.append(n_uav)
        for uav in tmp: self.uavs.append(UAV.Child(self.current,uav,self.drone_number))

#   def getKey(item):
#       return item[0]

    def fit(self,unservice_set, threshold, drone_limit, k, sample):
        from operator import itemgetter
        from functools import reduce
        import numpy as np
        '''
        Performs a fitness procedure on the UAV set.
        It will choose the best 50 Children to stay, and eliminate the rest objects.

        Parameters
        ----------
        unservice_set : [UE_set]
            The current iteration of the experiment
        threshold : (float)
            The threshold used to measure the fitness of the drones
        drone_limit : (int)
            The MAX number of chromosomes (paths) a child may have.
        k: (int)
            The number of clusters
```

```
        Returns
        -------

        '''

        distances, angle_ratios, intersections_set, services = [],[],[],[]
        distance, angle_ratio, intersections, service= 0,0,0,0
        for i,uav in enumerate(self.uavs):
            #[distance, angle_ratio, intersections, service] = uav.fitness(unservice_set, threshold)
            [distance, angle_ratio, intersections] = uav.fitness(unservice_set, threshold)
            distances.append(distance)
            intersections_set.append(intersections)
            angle_ratios.append(angle_ratio)
            #services.append(service)
            #print('--> Ratio: ',str(i*100/len(self.uavs)),' - ',i,' - ',len(self.uavs))
            #print(distance, angle_ratio, service)

        min_intersections = min(intersections_set)
        min_distance = min(distances)

        self.uav_list = []
        ###print("\n-->Eliminating extra childs")
        ##ponderate and sort all uav childs by best fit

        #for distance_total,angle_ratio, path_intersections,service_ratio,uav in
zip(distances,angle_ratios,services, intersections_set,self.uavs):
        for distance_total,angle_ratio, path_intersections,uav in zip(distances,angle_ratios,
intersections_set,self.uavs):
            if(path_intersections != 0):
                intersection_ratio = min_intersections/path_intersections
            else:
                intersection_ratio = 1

            distance_ratio = min_distance/distance_total
            #grade = service_ratio*0.5 + distance_ratio*0.25 + angle_ratio*0.25
            score = distance_ratio*0.3 + angle_ratio*0.4 + intersection_ratio*0.3
            l1=[]
            for path in [x for x in uav.chromosomes]:
                for point in path:
                    l1.append(point)
            tot_k = reduce(lambda x,y: x+y, [len(x) for x in uav.chromosomes])
            #Filter children with more drones than expected
            if((len(uav.chromosomes)>drone_limit) or (tot_k != k) or (len(l1) !=
len(np.unique(l1,axis=0)))):
                self.uav_list.append([0,uav])
            else:
                self.uav_list.append([score,uav])
```

```
        #delete not wanted childs [object wise]

        self.uav_list = sorted(self.uav_list, key=itemgetter(0), reverse=True)
        for uav in self.uav_list[sample:]:
            del(uav[1])

        self.uav_list = list(filter(lambda x: len(x)>1, self.uav_list))
        self.uavs = [uav[1] for uav in self.uav_list]
        self.best_uav = self.uav_list[0]

def graph_path(specimen_candidates, unserviced_points, unserviced_centroids, loss, drones):
    '''
    Creates graphs from the evolution of candidates

    Parameters
    ----------
    specimen_candidates: {candidates, score}
        A dictionary that contains candidates and their scores

    unserviced_points: [point]
        A set of (points) with unserviced UE

    loss: (int)
        BS percentage loss for the set

    drones: (int)
        Number of drones allowed for he set

    Returns
    ----------
    None

    '''
    pic_p = 1
    fig_p = 1
    for i,drone in enumerate(specimen_candidates):
        fig1 = plt.figure(fig_p)
        plt.subplot(2,2,pic_p)
        pic_p += 1
        if(pic_p > 4):
            pic_p = 1
            fig_p += 1
        plt.plot([x[0] for x in unserviced_centroids],[x[1] for x in
unserviced_centroids],'g*',markersize=10)
        plt.plot([x[0] for x in unserviced_points],[x[1] for x in unserviced_points],'r.',markersize=0.5)
        path = drone['specimen'].chromosomes
        co=['y--','c--','m--','b--','k--','y-','c-','m-','b-','k-']
        for pa in path:
```

```
            x = [x[0] for x in pa]
            x.insert(0,x[-1])
            y = [x[1] for x in pa]
            y.insert(0,y[-1])
            plt.plot(x,y,co.pop(),linewidth=1)
        plt.xlabel('Km')
        plt.ylabel('Km')
        plt.title('Specimen %d with a score of %g %s, with a %d %s of BS loss, and %d drone(s)
allowed' % (i, drone['score']*100,'%', loss,'%', drones))
    fig1.show()

if __name__ == "__main__":
    #Create set experiment objects
    uep = UserEquipmentSet(100,'p',25)
    uen = UserEquipmentSet(100,'n',25)
    ueu = UserEquipmentSet(100,'u',25)
    bsp = MacroBaseSet(10,'p',25)
    bsn = MacroBaseSet(10,'n',25)
    bsu = MacroBaseSet(10,'u',25)

#    exp_pp = Experiment(uep, bsp, 200000)
#    exp_pp.process_loss()
#    exp_pp.process_uav(100)
#
#    exp_pn = Experiment(uep, bsn, 200000)
#    exp_pn.process_loss()
#    exp_pn.process_uav(100)
#
#    exp_pu = Experiment(uep, bsu, 200000)
#    exp_pu.process_loss()
#    exp_pu.process_uav(100)
#
#    exp_nn = Experiment(uen, bsn, 200000)
#    exp_nn.process_loss()
#    exp_nn.process_uav(100)
#
#    exp_np = Experiment(uen, bsp, 200000)
#    exp_np.process_loss()
#    exp_np.process_uav(100)
#
#    exp_nu = Experiment(uen, bsu, 200000)
#    exp_nu.process_loss()
#    exp_nu.process_uav(100)
#
#    exp_uu = Experiment(ueu, bsu, 200000)
#    exp_uu.process_loss()
#    exp_uu.process_uav(100)
```

```
    exp_up = Experiment(ueu, bsp, 200000)
    exp_up.process_loss()
    exp_up.process_uav(100)

    exp_un = Experiment(ueu, bsn, 200000)
    exp_un.process_loss()
    exp_un.process_uav(100)

            ##Code below used solely for testing
    #vor = Voronoi(list(zip([x for x in bs1.xbs], [y for y in bs1.ybs])))
    #fig = voronoi_plot_2d(vor, show_vertices=False, line_colors='black',line_width=1,
line_alpha=1, point_size=2)
    #plt.plot(ue1.xue,ue1.yue,'r.',bs1.xbs, bs1.ybs,'b^')
    #plt.show()

    element = -2
    loss1 = -1

    #graph_path(candidates, exp_pp.unserviced[element])

graph_path(exp_pp.results[element]['candidates'],exp_pp.unserviced[loss1],exp_pp.unserviced
_centroids[loss1], exp_pp.results[element]['loss'], exp_pp.results[element]['allowed_drones'])
    print('Ready!')
    #used to test Mutators.py
    #uav = uav_set1.uavs[0]
    #m_chromo = uav.chromosomes
    #chromo = m_chromo[0]

    #used for ploting
    #plt.plot([x[0] for x in exp1.unserviced[0]],[x[1] for x in exp1.unserviced[0]],'r.',[x[0] for x in
k],[x[1] for x in k],'g*')
```

UAV.py

```
# -*- coding: utf-8 -*-
"""
Performs UAV positioning and calculations through GA

@author: Jose Matamoros
"""
import numpy as np
import Mutators
import Fitness

class Child:
    '''
    Each class must be created for each child.
    Each child will have multiple chromosomes within, which will allow it to mutate.
    '''
    def __init__(self, current, centroids, drone_limit):
        '''
        Parameters
        ----------
        current : (int)
            The current iteration of the experiment

        centroids : []
            If current == 0 then it must be a set of cluster centroids
            else
            It must be a set of chromosomes
        '''
        self.born = current
        self.current = current
        if(current==0):
            self.chromosomes = Mutators.initial_build(centroids, drone_limit)
        else:
            self.chromosomes = centroids

    def mutate(self,current):
        '''
        Mutates and produces new children from the Child's chromosomes.

        Parameters
        ----------
        current : (int)
            The current iteration of the experiment

        '''
        self.current = current
        new_childs = []
```

```
        for new_ch in Mutators.mutator_in_chromo(self.chromosomes):
new_childs.append(new_ch)
        #for new_ch in Mutators.mutator_in_chromo(self.chromosomes):
new_childs.append(new_ch)
        if(len(self.chromosomes)>2):
            #for new_ch in Mutators.mutator_cross_chromo(self.chromosomes):
new_childs.append(new_ch)
            #for new_ch in Mutators.mutator_cross_chromo(self.chromosomes):
new_childs.append(new_ch)
            for new_ch in Mutators.mutator_cross_chromo(self.chromosomes):
new_childs.append(new_ch)

        return(new_childs)

    def fitness(self,unserviced_set, threshold):
        '''
        Parameters
        ----------
        unserviced_set : [UE]
            A set of UE that are in need of service

        threshold : (float)
            The threshold to meassure service

        Returns
        -------
        [float,float,float]
            Return he total distance of the specimen's paths, the angle ratio of all paths, and a ratio
of how many did it serviced

        '''

        distance = Fitness.total_distance(self.chromosomes)
        angle_ratio = Fitness.angle_ratio(self.chromosomes)
        intersections = Fitness.line_cross(self.chromosomes)
        #service = Fitness.service_ratio(self.chromosomes, unserviced_set.copy(), threshold)

        #return([distance, angle_ratio, intersections, service])
        return([distance, angle_ratio, intersections])

if __name__ == "__main__":
            ##Code below used solely for testing
    from ClusterCentroid import centroids
    #plt.plot(chromo[0][0], chromo[0][1], 'k-')

    k=centroids([x[0] for x in exp1.unserviced[0]],[x[1] for x in exp1.unserviced[0]])
    child1 = Child(0,k)
```

```
exp_chromo = initial_build(k, 5)
chromosomes = exp_chromo[0]
```

ReceivedPower.py

```
# -*- coding: utf-8 -*-
"""
Received Power Calculator

@author: Jose Matamoros
"""
from math import log

def receivedPower(pt,d,hb):
    '''
    Calculates the received power based on the Okamura-Hata model

    Parameters
    ----------
    pt : float
        Transmitted Power[dB]
    d : float
        Distance[m]
    hb: float
        Base Station height[m]

    Returns
    -------
    float
        Received Power [dB]

    '''

    f = 700 #Carrier Frequency = 700 MHz
    hm = 2  #Mobile height set to 2 meters

    am = (1.1*log(f,10)-0.7)*hm - (1.56*log(f,10)-0.8)
    a = 69.55 + 26.16*log(f,10) - 13.82*log(hb,10) - am
    b = 44.9 - 6.55*log(hb,10)
    c = 0
    pl = a + b*log(d,10) + c #Path loss calculation


    #pl = (44.9-6.55*log(hb,10))*log(d,10)+5.83*log(hb,10)+16.33 + 26.16*log(f,10)

    return (pt - pl) #Return Received Power = Transmitted Power - Path Loss

if __name__ == "__main__":
            ##Code below used solely for testing
    print(receivedPower(0, 0.04, 2))
```

Mutators.py

```
# -*- coding: utf-8 -*-
"""
Functions for child mutators

@author: Jose Matamoros
"""
import numpy as np

def initial_build(centroids,drone_limit):
    '''
    Generates a vecotr of 3 Vertice, with a graph 3+-2.
    A Graph(V,E) will now be called chromosome

    Parameters
    ----------
    centroids : [float,float]
        A position vector of centroids

    Returns
    -------
    [[int]]
        A matriz, of 1xn, where each n element is a chromosome
    '''
    from sklearn.utils import shuffle
    centroids = shuffle(centroids, random_state=0)
    div = round(len(centroids)/drone_limit + 0.0001 - 0.5)
    if(div<3):
        div = 3
    n=round(len(centroids)/div + 0.0001 - 0.5)
    chromosome = [[] for x in range(0,n)]
    loop = 0
    for point in centroids:
        chromosome[loop].append(point)
        loop+=1
        if(loop>=n):loop=0
    return(chromosome)

def mutator_in_chromo(chromosomes):
    '''
    Generates 3 chromosome sets from a given chromosome through 3 In-Chromosome methods

    Parameters
    ----------
    chromosomes : [chromosome]
        A set of chromosomes (Graph(V,E))
```

```
Returns
-------
3x[chromosomes]
   Where each element will be the provided set minus n selected chromosomes,
   and atached will be n mutatd chromosomes
'''
new_chromo=[]

#First In-Chromosome mutator
#Swap a section of the chromosome
tmp_chromosome = chromosomes.copy()
nl = len(tmp_chromosome)
s = int(round(np.random.uniform(0,nl-1,1)[0])-0.5)
chromosome = tmp_chromosome[s]
n=len(chromosome)
ini = int(round(np.random.uniform(0,n-3,1)[0]))
end = int(round(np.random.uniform(ini,n-1,1)[0]))
chromo1 = chromosome.copy()
chromo1[ini:end] = chromosome[ini:end][::-1]
#tmp_chromosome.tolist()
tmp_chromosome.pop(s)
tmp_chromosome.append(chromo1)
new_chromo.append(tmp_chromosome)

#Second In-Chromosome mutator
#Swaps the positions of two chromosomes
tmp_chromosome = chromosomes.copy()
nl = len(tmp_chromosome)
s = int(round(np.random.uniform(0,nl-1,1)[0])-0.5)
chromosome = tmp_chromosome[s]
n=len(chromosome)
pos = [int(round(x)) for x in np.random.uniform(0,n-1,2)]
chromo2 = chromosome.copy()
chromo2[pos[0]],chromo2[pos[1]] = chromosome[pos[1]],chromosome[pos[0]]
#tmp_chromosome.tolist()
tmp_chromosome.pop(s)
tmp_chromosome.append(chromo2)
new_chromo.append(tmp_chromosome)

#Third In-Chromosome mutator
#Inserts a chromosome in a different place
tmp_chromosome = chromosomes.copy()
nl = len(tmp_chromosome)
s = int(round(np.random.uniform(0,nl-1,1)[0])-0.5)
chromosome = tmp_chromosome[s]
n=len(chromosome)
pos = [int(round(x)) for x in np.random.uniform(0,n-1,2)]
chromo3 = chromosome.copy()
```

```
    chromo3.pop(pos[0])
    chromo3.insert(pos[1],chromosome[pos[0]])
    #tmp_chromosome.tolist()
    tmp_chromosome.pop(s)
    tmp_chromosome.append(chromo3)
    new_chromo.append(tmp_chromosome)

    return([x for x in new_chromo])

def mutator_cross_chromo(chromosomes):
    '''
    Generates 4 new childs from a given chromosome thorugh 3 Cross-Chromosome methods

    Parameters
    ----------
    chromosomea : [chromosome]
        A set of chromosomes (Graph(V,E))

    Returns
    -------
    kx[chromosomes]
        A vector of size up to 4 new set of chromosomes
    '''

    new_chromo=[]
    #First Cross-Chromo mutator
    #Vertex swapping between chromosomes

    tmp_chromosome = chromosomes.copy()
    n = len(chromosomes)
    pos = [int(round(np.random.uniform(0,n-1,1)[0])-0.5),int(round(np.random.uniform(0,n-
1,1)[0])-0.5)]
    while(pos[0] == pos[1]): pos[1] = int(round(np.random.uniform(0,n-1,1)[0]))
    chromo1, chromo2 = chromosomes[pos[0]],chromosomes[pos[1]]
    n1,n2 = len(chromo1),len(chromo2)
    ini1 = int(round(np.random.uniform(0,n1-2,1)[0]))
    end1 = int(round(np.random.uniform(ini1+1,n1-1,1)[0]))
    ini2 = int(round(np.random.uniform(0,n2-2,1)[0]))
    end2 = int(round(np.random.uniform(ini2+1,n2-1,1)[0]))
    tmp1 = []
    tmp1.extend(chromo1[:ini1])
    tmp2 = []
    tmp2.extend(chromo2[:ini2])
    aux1=chromo1[ini1:end1]
    aux2=chromo2[ini2:end2]
    for x in aux2: tmp1.append(x)
    for x in aux1: tmp2.append(x)
    for x in chromo1[end1:]: tmp1.append(x)
```

```
for x in chromo2[end2:]: tmp2.append(x)
if((len(tmp1)>2) and (len(tmp2)>2)):
   #tmp_chromosome.tolist()
   tmp_chromosome.pop(pos[0])
   if(pos[0]<pos[1]):
      pos[1]-=1
   tmp_chromosome.pop(pos[1])
   tmp_chromosome.append(tmp1)
   new_chromo.append(tmp_chromosome)
   tmp_chromosome.append(tmp2)
   new_chromo.append(tmp_chromosome)

'''
for i,chromo1 in enumerate(chromosomes):
   n1 = len(chromo1)
   for j,chromo2 in enumerate(chromosomes):
      if(j!=i):
         n2 = len(chromo2)
         ini1 = int(round(np.random.uniform(0,n1-1,1)[0]))
         end1 = int(round(np.random.uniform(ini1,n1-1,1)[0]))
         ini2 = int(round(np.random.uniform(0,n2-1,1)[0]))
         end2 = int(round(np.random.uniform(ini2,n2-1,1)[0]))
         tmp1 = []
         tmp1.extend(chromo1[:ini1])
         tmp2 = []
         tmp2.extend(chromo2[:ini2])
         aux1=chromo1[ini1:end1]
         aux2=chromo2[ini2:end2]
         for x in aux2: tmp1.append(x)
         for x in aux1: tmp2.append(x)
         for x in chromo1[end1:]: tmp1.append(x)
         for x in chromo2[end2:]: tmp2.append(x)
         new_chromo.append(tmp1)
         new_chromo.append(tmp2)
'''

#Second Cross-Chromo mutator
#Joines two chromosomes
tmp_chromosome = chromosomes.copy()
n = len(chromosomes)
if(n > 1):
   f = int(round(np.random.uniform(0,n-1,1)[0])-0.5)
   e = f
   while(e == f): e = int(round(np.random.uniform(0,n-1,1)[0])-0.5)
   tmp = tmp_chromosome[f][:]
   for x in tmp_chromosome[e]: tmp.append(x)
   #tmp_chromosome.tolist()
   tmp_chromosome.pop(e)
```

```
        tmp_chromosome.pop(f)
        tmp_chromosome.append(tmp)
        new_chromo.append(tmp_chromosome)

    #Third Cross-Chromo mutator
    #Separates a chromosome into two chromosomes
    tmp_chromosome = chromosomes.copy()
    s = int(round(np.random.uniform(0,n-1,1)[0])-0.5)
    chromo = chromosomes[s]
    n1 = len(chromo)
    if(n1>=6):
        ini = int(round(np.random.uniform(0,n1-4,1)[0]))
        end = int(round(np.random.uniform(ini,n1-1,1)[0]))
        while(end == ini or end<ini+3): end = int(round(np.random.uniform(ini,n1-1,1)[0]))
        tmp1 = chromo[ini:end]
        tmp2 = chromo[:ini]
        for x in chromo[end:]: tmp2.append(x)
        #tmp_chromosome.tolist()
        if(len(tmp1)>2 and len(tmp2)>2):
            tmp_chromosome.pop(s)
            tmp_chromosome.append(tmp1)
            tmp_chromosome.append(tmp2)
            new_chromo.append(tmp_chromosome)

    return(new_chromo)

if __name__ == "__main__":
            ##Code below used solely for testing
    next
    #chromosomes = m_chromo.copy()
    #new_chromo = []
```

Fitness.py

```
# -*- coding: utf-8 -*-
"""
Fitness Functions

@author: Jose Matamoros
"""
from AssignedBS import distance

def ccw(A,B,C):
    return (C[1]-A[1]) * (B[0]-A[0]) > (B[1]-A[1]) * (C[0]-A[0])

# Return true if line segments AB and CD intersect
def intersect(A,B,C,D):
    return ccw(A,C,D) != ccw(B,C,D) and ccw(A,B,C) != ccw(A,B,D)

def line_cross(chromosome_set):
    '''
    The ratio of intersected paths the specimen has

    Parameters
    ----------
    chromosome_set : [chromosome]
        A set of chromosomes

    Returns
    -------
    (int)
        the number of intersections between all paths within the specimen
    '''

    paths = []
    for path in chromosome_set[:]:
        tmp_lines = path[:]
        tmp_lines.append(path[0])
        paths.append(tmp_lines)

    segments=[]
    for path in paths:
        aa = [point for point in path][:-1]
        bb = [point for point in path][1:]
        tmp_segments = []
        for point_a,point_b in zip(aa,bb):
            tmp_segments.append([point_a,point_b])
        segments.append(tmp_segments)

    intersections = 0
```

```
    for i,lines in enumerate(segments):
        for j,lines2 in enumerate(segments):
            if(j>i):
                for line in lines:
                    for line2 in lines2:
                        if(intersect(line[0],line[1],line2[0],line2[1])):
                            intersections += 1
    return(intersections)

def total_distance(chromosome_set):
    '''
    The total distance of all the paths for a given child (set of chromosomes)

    Parameters
    ----------
    chromosome_set : [chromosome]
        A set of chromosomes

    Returns
    -------
    (float)
        The sum of all the distances of all paths (chromosomes)
    '''
    import numpy as np

    suma=0
    for path in chromosome_set.copy():
        aa = [point for point in path][:-1]
        bb = [point for point in path][1:]
        for point_a, point_b in zip(aa,bb):
            suma += distance(point_a,point_b)
        suma += distance(aa[0],bb[-1])
    return(suma)

def angle(a,b,c):
    '''
    Returns the angle ABC

    Parameters
    ----------
    a,b,c : [point,poin,point]
        Point (X,Y)

    Returns
    -------
    (float)
        the angle ABC
    '''
```

```
    from math import acos

    ab = distance(a,b)
    bc = distance(b,c)
    ca = distance(c,a)
    return( acos((ab**2 + bc**2 - ca**2)/(2*ab*bc)) )

def angle_ratio(chromosome_set):
    '''
    Returns a ratio of how much does the shape is similar to a perfect polygon
    with the ame number of angles

    Parameters
    ----------
    chromosome_set : [chromosome]
        The current iteration of the experiment

    Returns
    -------
    (float)
        The total calculated ratio of angles from all individual angles
        compared to an angle from a perfect polygon of n vertex
    '''
    from math import pi
    from functools import reduce

    ratio = []
    for path in chromosome_set.copy():
        points = [point for point in path]
        n = len(points)
        perfect_angle = (n-2)*pi/n
        points.insert(0,points[-1])
        points.append(points[0])
        suma = 0
        for i in range(1,n):
            tmp_angle = angle(points[i-1],points[i],points[i+1])
            if(tmp_angle>pi): tmp_angle = pi - tmp_angle
            suma += abs((tmp_angle - pi)/pi)
        ratio.append(1 - suma/n)
    return (reduce((lambda x,y: x+y), ratio)/len(chromosome_set))

def service_ratio(chromosome_set, unserviced_set, threshold):
    '''
    Returns a ratio which provides an idea of how users are serviced.
    In addition, it modifies the unserviced_set list provided by name bonding

    Parameters
    ----------
```

```
    chromosome_set : [chromosome]
        The current iteration of the experiment
    unserviced_set: [points]
        A set of unserviced UE
    threshold: (float)
        The usage threshold from the experiment

    Returns
    -------
    (float)
        A ratio of temporarly serviced UE with the solution
    '''
    from AssignedBS import distance
    from CapacityCal import capacity
    from AssignedBS import assignedUAV

    unserviced_num = len(unserviced_set)

    diameter = 3
    points = []
    for path_i in chromosome_set:
        ##Calculate new points in path with the given radius
        path = path_i.copy()
        n=len(path)
        path.append(path[0])
        for i in range(0,n): #each line
            magnitude = distance(path[i],path[i+1])
            points.append(path[i])
            div = round(magnitude/diameter - 0.5)
            if(div > 3):
                for divisor in range(1,round(magnitude - 0.5),diameter):
                    t = divisor/magnitude
                    xt,yt = [(1-t)*path[i][0]+t*path[i+1][0],(1-t)*path[i][1]+t*path[i+1][1]]
                    points.append([xt,yt])
            points.append(path[-1])

            ##Calculate if UE is being serviced during the path
            capacities = []
            capacities =
capacity(unserviced_set,points,assignedUAV(unserviced_set,points),threshold)
            for i,cap in enumerate(capacities):
                if(cap[1]>0):
                    unserviced_set.pop(i)

    ##HERE YOU MUST CALCULATE HOW MANY WERE LEFT WITHOUT SERVICE
    return(len(unserviced_set)/unserviced_num)

if __name__ == "__main__":
```

```
        ##Code below used solely for testing
next
#plt.plot([x[0] for x in points],[x[1] for x in points],'r.')
#fig1.show()
dd = uv_list[0][1].chromosomes
paths = []
for path in dd[:]:
   tmp_lines = path[:]
   tmp_lines.append(path[0])
   paths.append(tmp_lines)

segments=[]
for path in paths:
   aa = [point for point in path][:-1]
   bb = [point for point in path][1:]
   tmp_segments = []
   for point_a,point_b in zip(aa,bb):
      tmp_segments.append([point_a,point_b])
   segments.append(tmp_segments)

intersections = 0
for i,lines in enumerate(segments):
   for j,lines2 in enumerate(segments):
      if(j>i):
         for line in lines:
            for line2 in lines2:
               if(intersect(line[0],line[1],line2[0],line2[1])):
                  intersections += 1
```

ClusterCentroid.py

```
# -*- coding: utf-8 -*-
"""
K-means Cluster Calculator

@author: Jose Matamoros
"""
from sklearn.cluster import AffinityPropagation
import numpy as np
def centroids(xue, yue):
    '''
    Provides the centroids for a given cluster

    Parameters
    ----------
    xue : [float]
        Vector with X positions for the set
    yue : float
        Vector with Y positions for the set

    Returns
    -------
    [float]
        A Vector with X and Y positions of the centroids.

    '''
    arr = []
    for x, y in zip(xue, yue):
        arr.append([x,y])

    arr = np.array(arr)
    clustering = AffinityPropagation(damping=0.60).fit(arr)
    k = clustering.cluster_centers_
    return(k)

def centroids_adv(xue, yue):
    '''
    Provides the centroids for a given cluster

    Parameters
    ----------
    xue : [float]
        Vector with X positions for the set
    yue : float
        Vector with Y positions for the set

    Returns
```

```
    -------
    [float]
        A Vector with X and Y positions of the centroids.

    '''
    arr = []
    for x, y in zip(xue, yue):
        arr.append([x,y])

    arr = np.array(arr)
    clustering = AffinityPropagation().fit(arr)
    k = clustering.cluster_centers_
    return(k)

if __name__ == "__main__":
            ##Code below used solely for testing
    import matplotlib.pyplot as plt
    import csv

#    with open('Xue.csv') as csvfile:
#        r = csv.reader(csvfile, delimiter=',')
#        xue = [float(ele) for ele in ([rr for rr in r][0])]
#
#    with open('Yue.csv') as csvfile:
#        r = csv.reader(csvfile, delimiter=',')
#        yue = [float(ele) for ele in ([rr for rr in r][0])]

    numue = np.random.poisson(2500, 1)
    xue = np.random.uniform(0,25,numue)
    yue = np.random.uniform(0,25,numue)

    k=centroids(xue, yue)
    #k2=centroids_adv(xue, yue)
    k2 = centroids_adv([x[0] for x in k],[x[1] for x in k])

    plt.plot(xue, yue, 'r.',[kk[0] for kk in k],[kk[1] for kk in k],'b^')
    plt.show()
    plt.plot(xue, yue, 'r.',[kk[0] for kk in k2],[kk[1] for kk in k2],'b^')
    plt.show()
```

CapacityCal.py

```
# -*- coding: utf-8 -*-
"""
Calculates the capacity (bps) of a UE based on SNR

@author: Jose Matamoros
"""

from ReceivedPower import receivedPower
from AssignedBS import distance
from math import log
from math import sqrt

def db_to_w(power_db):
    return(10**(power_db/10))

def capacity(ue_set, bs_set, assigned_bs_list, capacity_threshold):
    '''
    Calculate each UE capacity based on the Rx from all BS

    Parameters
    ----------
    ue_set : float[]
        UE set object that includes all positions
    bs_set : float
        BS set object that includes all positions
    assigned_bs_list : float[]
        BS index and power assigne in order of each UE
    capacity_threshold : float
        The capacity threshold in bps

    Returns
    -------
    [float,int]
        a 2xn matrix with [0] as the capacity in bps, and [1] as a BOOLEAN whether is within the
threshold
    '''
    bw=1.4e6
    ue_capacity = []
    for ue,a_bs in zip(ue_set,assigned_bs_list):
        noise = 0
        for i,bs in enumerate(bs_set):
            if(i!=a_bs[0]):
                #with open('test.txt', 'a') as of: of.write("poins: " + str(ue) +" - "+ str(bs)+'\n')
                try:
                    noise += db_to_w(receivedPower(30,distance(ue,bs),20))**2
                except:
```

```
            print('points: ',ue,bs)
        snr = abs(db_to_w(a_bs[1])/sqrt(noise/(len(bs_set)-1)))
        c= bw*log(1+snr,10)
        if ((c>= capacity_threshold)and(a_bs[1] > -102)):
            ue_capacity.append([c,1])
        else:
            ue_capacity.append([c,0])
    return(ue_capacity)

def capacityBS_from_UE(ue_set, bs_set, assigned_ue_list, capacity_threshold):
    ##Currently unsed !
    '''
    Calculate each UE capacity based on the Rx from all BS

    Parameters
    ----------
    ue_set : float[]
        UE set object that includes all positions
    bs_set : float
        BS set object that includes all positions
    assigned_ue_list : float[]
        BS index and power assigne in order wit evey UE that it services
    capacity_threshold : float
        The capacity threshold in bps

    Returns
    -------/
    [float,int]
        a 2xn matrix with [0] as the capacity in bps for each user, and [1] as a BOOLEAN whether is
within the threshold

    '''
    bw=1.4e6
    bs_to_ue_capacity = [[] for i in range(0,len(ue_set))]
    for bs,bs_pos in zip(assigned_ue_list,bs_set):
        for i,ue in enumerate(bs):
            if(len(bs)!=1):
                noise=0
            else:
                noise=1
            for j,ue_extra in enumerate(bs):
                if(i!=j):
                    noise += db_to_w(receivedPower(-10,distance(ue_extra,bs_pos),20))**2
            if(len(bs)!=1):
                snr = abs(db_to_w(ue[1])/sqrt(noise/(len(bs)-1)))
            else:
                noise = -112
                snr = abs(db_to_w(ue[1])/noise)
```

```
        c= bw*log(1+snr,10)
        if ((c>= capacity_threshold)and(ue[1] > -102)):
            bs_to_ue_capacity[ue[0]] = [c,1]
        else:
            bs_to_ue_capacity[ue[0]] = [c,0]
    return(bs_to_ue_capacity)


if __name__ == "__main__":
        ##Code below used solely for testing
    from AssignedBS import assignedBS
    from AssignedBS import assignedUE
    #capacity(zip([x for x in ue1.xue], [y for y in ue1.yue]),zip([x for x in bs1.xbs], [y for y in
bs1.ybs]),assi,500e3)
    ue_set = list(zip([x for x in ue1.xue], [y for y in ue1.yue]))
    bs_set = list(zip([x for x in bs1.xbs], [y for y in bs1.ybs]))
    #bs_set = bs_set[:round(len(bs_set)*0.1)]
    #plt.plot([x[0] for x in ue_set],[x[1] for x in ue_set],'r.',[x[0] for x in bs_set],[x[1] for x in
bs_set],'b^')
    assigned_bs_list=assignedBS(ue_set,bs_set)
    assigned_ue_list=assignedUE(ue_set,bs_set)
    bw=1.4e6
    capacity_threshold = 200000
    ue_capacity = []
    for ue,a_bs in zip(ue_set,assigned_bs_list):
        noise = 0
        for i,bs in enumerate(bs_set):
            if(i!=a_bs[0]):
                noise += db_to_w(receivedPower(30,distance(ue,bs),20))**2
        snr = abs(db_to_w(a_bs[1])/sqrt(noise/(len(bs_set)-1)))
        c= bw*log(1+snr,10)
        if ((c>= capacity_threshold)and(a_bs[1] > -102)):
            ue_capacity.append([c,1])
        else:
            ue_capacity.append([c,0])
```

AssignedBS.py

```
# -*- coding: utf-8 -*-
"""
Claculates each UE assigned BS

@author: Jose Matamoros
"""
from math import sqrt
from ReceivedPower import receivedPower

def distance(a,b):
    '''
    Calculate the distance between points a and b

    Parameters
    ----------
    a : [float,float]
        A set of points (x,y)

    b : [float,float]
        A set of points (x,y)

    Returns
    -------
    float
        Distance between a and b

    '''
    c = ((a[0]-b[0]) ** 2) + ((a[1]-b[1])**2)
    d = sqrt(c)
    if (d<0.001): d = 0.001
    return(d)

def distanceUAV(a,b):
    '''
    Calculate the distance between a and an Aerial Base Station b

    Parameters
    ----------
    a : [float,float]
        A set of points (x,y)

    b : [float,float]
        A set of points (x,y)

    Returns
    -------
```

```
    float
        Distance between a and b

    '''
    c = ((a[0]-b[0]) ** 2) + ((a[1]-b[1])**2) + ((0 - 0.05)**2)
    return(sqrt(c))

def assignedBS(ue_set,bs_set):
    '''
    Calculate the assigned BS fpr each UE

    Parameters
    ----------
    ue_set : float
        UE set object that includes all positions
    bs_set : float
        BS set object that includes all positions

    Returns
    -------
    [int,float]
        An oredered vector. Where the first value will be for the first UE,
        and its value will be the vector position of the BS
        [2,    BS2    [UE1,R_PWR]
        4,     BS4    [UE2,R_PWR]
        3,     BS3    [UE3,R_PWR]
        2]     BS2    [UE4,R_PWR]

    '''
    assigned = []
    for ue in ue_set:
        as_i=0
        Pt = 30
        as_pow=-1000
        for i,bs in enumerate(bs_set):
            temp_power = receivedPower(Pt,distance(ue,bs),20)
            if(temp_power>as_pow):
                as_pow = temp_power
                as_i = i
        assigned.append([as_i,as_pow])
    return(assigned)

def assignedUAV(ue_set,bs_set):
    '''
    Calculate the assigned BS fpr each UAV

    Parameters
    ----------
```

```
    ue_set : float
        UE set object that includes all positions
    bs_set : float
        UAV BS set object that includes all positions

    Returns
    -------
    [int,float]
        An oredered vector. Where the first value will be for the first UE,
        and its value will be the vector position of the BS
        [2,    BS2    [UE1,R_PWR]
        4,     BS4    [UE2,R_PWR]
        3,     BS3    [UE3,R_PWR]
        2]     BS2    [UE4,R_PWR]

    '''
    assigned = []
    for ue in ue_set:
        as_i=0
        Pt = 20
        as_pow=-1000
        for i,bs in enumerate(bs_set):
            temp_power = receivedPower(Pt,distanceUAV(ue,bs),20)
            if(temp_power>as_pow):
                as_pow = temp_power
                as_i = i
        assigned.append([as_i,as_pow])
    return(assigned)

def assignedUE(ue_set,bs_set):
    '''
    Calculate the assigned UE for each BS

    Parameters
    ----------
    ue_set : float
        UE set object that includes all positions
    bs_set : float
        BS set object that includes all positions

    Returns
    -------
    [int,float]
        An oredered matrix, where the position represents he BS, and within there will be a list that
represents all UE assigned to it.

        [[1,8,32]          BS1    [UE1,R_PWR],[UE8,R_PWR],[UE32,R_PWR]
```

```
        [2,5,10,13,14],      BS2
[UE2,R_PWR],[UE5,R_PWR],[UE10,R_PWR],[UE13,R_PWR],[UE14,R_PWR]
        [3,4,16,21]]         BS3    [UE3,R_PWR],[UE4,R_PWR],[UE16,R_PWR],[UE21,R_PWR]

    '''
    assigned = [[] for i in range(0,len(bs_set))]
    for u,ue in enumerate(ue_set):
        as_i=0
        as_pow=-1000
        Pt = 30
        for i,bs in enumerate(bs_set):
            temp_power = receivedPower(Pt,distanceUAV(ue,bs),20)
            if(temp_power>as_pow):
                as_pow = temp_power
                as_i = i
        assigned[as_i].append([u,as_pow])
    return(assigned)

if __name__ == "__main__":
            ##Code below used solely for testing
    ue_set = list(zip([x for x in ue1.xue], [y for y in ue1.yue]))
    bs_set = list(zip([x for x in bs1.xbs], [y for y in bs1.ybs]))
    UE_ass = assignedUE(ue_set,bs_set)
```

Graphics.py

```
"""
The execution of this file would create joined result graphs provided that the simulations have
been done.

"""
'''
for n in range(0,len(exp1.unserviced)):
   k=centroids([x[0] for x in exp1.unserviced[n]],[x[1] for x in exp1.unserviced[n]])
   fg1 = plt.plot([x[0] for x in exp1.unserviced[n]],[x[1] for x in
exp1.unserviced[n]],'r.',markersize=2)
   fg1 = plt.plot([x[0] for x in k],[x[1] for x in k],'g*',markersize=12)
   tot_l = len(bs1.xbs)
   BStempX = bs1.xbs[:round(tot_l*0.1*(n+1))]
   BStempY = bs1.ybs[:round(tot_l*0.1*(n+1))]
   fg1 = plt.plot(BStempX,BStempY,'b^',markersize=8)
   plt.show(fg1)

'''

'''



'''
from functools import reduce
import numpy as np
from scipy.ndimage.filters import gaussian_filter1d
from Fitness import *

#plt.hist([len(x['best_specimen']['specimen'].chromosomes) for x in exp_pp.results])


plt.hist([[len(x['best_specimen']['specimen'].chromosomes) for x in exp_pp.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_pn.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_pu.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_np.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_nn.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_nu.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_up.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_un.results],
        [len(x['best_specimen']['specimen'].chromosomes) for x in exp_uu.results],
        ], label=['UBPP', 'UBPN', 'UBPU','UBNP', 'UBNN', 'UBNU','UBUP', 'UBUN', 'UBUU'])
plt.xticks(np.arange(5), ('1', '2', '2', '3', '4'))
plt.autoscale(tight=True)
plt.grid('False')
plt.legend(loc='upper right')
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
```

```
plt.xlabel('NUMBER OF PATHS SELECTED FROM THE BEST SPECIMEN', fontsize='18')
plt.ylabel('NUMBER OF EXPERIMENTS', fontsize='18')
plt.title('NUMBER OF DETERMINED PATHS FOR ALL EXPERIMENTS FOR ALL DISTRIBUIONS FOR
UE AND BS', fontsize='18')

-----------------------------------------------

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_pp.results]))]) for k in range(10,100,10)], sigma=1),'-*b',markersize=12,
label='UBPP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_pn.results]))]) for k in range(10,100,10)], sigma=1),'-og',markersize=12,
label='UBPN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_pu.results]))]) for k in range(10,100,10)], sigma=1),'-^r',markersize=12,
label='UBPU')

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_np.results]))]) for k in range(10,100,10)], sigma=1),'-vc',markersize=12,
label='UBNP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_nn.results]))]) for k in range(10,100,10)], sigma=1),'-Dm',markersize=12,
label='UBNN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_nu.results]))]) for k in range(10,100,10)], sigma=1),'-sy',markersize=12,
label='UBNU')

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_up.results]))]) for k in range(10,100,10)], sigma=1),'-Pk',markersize=12,
label='UBUP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_un.results]))]) for k in range(10,100,10)], sigma=1),'-pb',markersize=12,
label='UBUN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_score'] for y in
list(filter(lambda x: x['loss']==k,
        [x for x in exp_uu.results]))]) for k in range(10,100,10)], sigma=1),'-Xg',markersize=12,
label='UBUU')

plt.legend()
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
```

```
plt.xlabel('BS loss [%]', fontsize='18')
plt.ylabel('MAXIMUM SCORE OVER 1', fontsize='18')
plt.title('MAXIMUM SCORE PER BS LOSS PERCENTAGE EXPERIMENTS WITH POISSON
DISTRIBUION FOR UE AND BS', fontsize='18')

-----------------------------------------------

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_pp.results]))]) for k in range(10,100,10)], sigma=1),'-*b',markersize=12,
label='UBPP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_pn.results]))]) for k in range(10,100,10)], sigma=1),'-og',markersize=12,
label='UBPN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_pu.results]))]) for k in range(10,100,10)], sigma=1),'-^r',markersize=12,
label='UBPU')

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_np.results]))]) for k in range(10,100,10)], sigma=1),'-vc',markersize=12,
label='UBNP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_nn.results]))]) for k in range(10,100,10)], sigma=1),'-Dm',markersize=12,
label='UBNN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_nu.results]))]) for k in range(10,100,10)], sigma=1),'-sy',markersize=12,
label='UBNU')

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_up.results]))]) for k in range(10,100,10)], sigma=1),'-Pk',markersize=12,
label='UBUP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_un.results]))]) for k in range(10,100,10)], sigma=1),'-pb',markersize=12,
label='UBUN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([len(y['candidates']) for y in
list(filter(lambda x: x['loss']==k,
      [x for x in exp_uu.results]))]) for k in range(10,100,10)], sigma=1),'-Xg',markersize=12,
label='UBUU')

plt.legend()
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
```

```
plt.xlabel('BS loss [%]', fontsize='18')
plt.grid('False')
plt.ylabel('NUMBER OF CANDIDATES FOR BEST SPECIMEN', fontsize='18')
plt.title('NUMBER OF CANDIDATES NEEDED TO FIND THE BEST SPECIMEN FOR EXPERIMENTS
WITH POISSON DISTRIBUION FOR UE AND BS', fontsize='18')

-----------------------------------------------

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_pp.results]))]) for k in range(10,100,10)], sigma=1),'-*b',markersize=12,
label='UBPP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_pn.results]))]) for k in range(10,100,10)], sigma=1),'-og',markersize=12,
label='UBPN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_pu.results]))]) for k in range(10,100,10)], sigma=1),'-^r',markersize=12,
label='UBPU')

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_np.results]))]) for k in range(10,100,10)], sigma=1),'-vc',markersize=12,
label='UBNP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_nn.results]))]) for k in range(10,100,10)], sigma=1),'-Dm',markersize=12,
label='UBNN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_nu.results]))]) for k in range(10,100,10)], sigma=1),'-sy',markersize=12,
label='UBNU')

plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_up.results]))]) for k in range(10,100,10)], sigma=1),'-Pk',markersize=12,
label='UBUP')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_un.results]))]) for k in range(10,100,10)], sigma=1),'-pb',markersize=12,
label='UBUN')
plt.plot([i for i in range(10,100,10)],gaussian_filter1d([max([y['best_specimen']['specimen'].born
for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_uu.results]))]) for k in range(10,100,10)], sigma=1),'-Xg',markersize=12,
label='UBUU')

plt.legend()
```

```
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.xlabel('BS loss [%]', fontsize='18')
plt.ylabel('Generations', fontsize='18')
plt.title('MAXIMUM NUMBER OF GENERATIONS NEEDED TO FIND BEST SPECIMEN WITH
POISSON DISTRIBUION FOR UE AND BS', fontsize='18')

-----------------------------------------------

vor = Voronoi(list(zip([x for x in bsn.xbs], [y for y in bsn.ybs])))
fig = voronoi_plot_2d(vor, show_vertices=False, line_colors='black',line_width=0.75,
line_alpha=0.75, point_size=1)
plt.plot(uen.xue,uen.yue,'r.',label='UE')
plt.plot(bsn.xbs, bsn.ybs,'b^', label='BS',markersize=7)
plt.xlabel('Km', fontsize='18')
plt.ylabel('Km', fontsize='18')
plt.legend()
plt.legend(loc='upper right')
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.title('Point placement and Voronoi diagram for UBNN', fontsize='18')
plt.show()

-----------------------------------------------
tot_l = len(exp_pp.BSset.xbs)
plt.plot(uep.xue, uep.yue, 'gP', label='With Service')
plt.plot([x[0] for x in exp_pp.unserviced[0]],[x[1] for x in exp_pp.unserviced[0]],'ro', label='No
Service')
##plt.plot(bsn.xbs[:round(tot_l*0.1)], bsn.ybs[:round(tot_l*0.1)], 'b^')
plt.xlabel('Km', fontsize='18')
plt.ylabel('Km', fontsize='18')
plt.legend(loc='upper right')
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.title('UBPP with 80% BS loss', fontsize='18')
plt.show()

-----------------------------------------------

element = 1
graph_path(exp_pp.results[element]['candidates'],exp_pp.unserviced[element],exp_pp.unservic
ed_centroids[element], exp_pp.results[element]['loss'],
exp_pp.results[element]['allowed_drones'])

exp_pu.showResults()

-----------------------------------------------
col = ['b','g','r','c','m','y','k','b','g']
ha = ('/','o','+','','v','D','p','P','//')
plt.hist([x[1] for x in exp_pp.initial_capacity],hatch='/', label='UBPP')
```

```
plt.hist([[x[1] for x in exp_pp.initial_capacity],
        [x[1] for x in exp_pn.initial_capacity],
        [x[1] for x in exp_pu.initial_capacity],
        [x[1] for x in exp_np.initial_capacity],
        [x[1] for x in exp_nn.initial_capacity],
        [x[1] for x in exp_nu.initial_capacity],
        [x[1] for x in exp_up.initial_capacity],
        [x[1] for x in exp_un.initial_capacity],
        [x[1] for x in exp_uu.initial_capacity],
        ], label=['UBPP', 'UBPN', 'UBPU','UBNP', 'UBNN', 'UBNU','UBUP', 'UBUN', 'UBUU'])

plt.autoscale(tight=True)
plt.legend(loc='upper left')
plt.grid(color='k', linestyle='-', linewidth=0.5)
plt.xlabel('Service', fontsize='18')
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.xticks([0,1],['No Service','With Service'], fontsize='18')
plt.ylabel('Ammount of UE', fontsize='18')
plt.title('Number of UE that have service in Initial Conditions', fontsize='18')

-----------------------------------------------

fig1 = plt.figure(1)

col = ['-*b','-og','-^r','-vc','-Dm','-sy','-Pk','-pb','-Xg']
lab = ['UBPP','UBPN','UBPU','UBNP','UBNN','UBNU','UBUP','UBUN','UBUU']

for j,k in enumerate((exp_pp,exp_pn,exp_pu,exp_np,exp_nn,exp_nu,exp_up,exp_un,exp_uu)):
   prt = []
   for cap in k.capacities:
      [a,b,c] = plt.hist([x[1] for x in cap],2)
      prt.append(a[0]*100/(a[0]+a[1]))
   fig2 = plt.figure(2)
   plt.plot([x for x in range(10,100,10)],[100-x for x in prt[::-1]],col[j], label=lab[j])
   fig1 = plt.figure(1)


fig2 = plt.figure(2)
plt.legend()
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.xlabel('BS loss [%]', fontsize='18')
plt.ylabel('UE serviced [%]', fontsize='18')
plt.title('UE WITH SERVICE AFTER BS LOSS', fontsize='18')
plt.grid(True)
fig2.show()

-----------------------------------------------
```

```
tot = []

col = ['-*b','-og','-^r','-vc','-Dm','-sy','-Pk','-pb','-Xg']
lab = ['UBPP','UBPN','UBPU','UBNP','UBNN','UBNU','UBUP','UBUN','UBUU']

for dis in (exp_pp,exp_pn,exp_pu,exp_np,exp_nn,exp_nu,exp_up,exp_un,exp_uu):
  cov_dis = []
  for loss in range(10,100,10):
     cov = []
     for itt in [x for x in dis.results]:
        if(itt['loss'] == loss):

cov.append(service_ratio(itt['best_specimen']['specimen'].chromosomes,dis.unserviced[int((loss
)/10)-1],200000))
     print('The coverage ratio is %g with a loss of %d'%(min(cov), loss))
     cov_dis.append(min(cov))
  plt.plot([i for i in range(10,100,10)],cov_dis)
  tot.append(cov_dis)

plt.hist(tot, label=['UBPP', 'UBPN', 'UBPU','UBNP', 'UBNN', 'UBNU','UBUP', 'UBUN', 'UBUU'])
plt.legend()
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.xlabel('BS loss [%]', fontsize='18')
plt.ylabel('UE serviced [%]', fontsize='18')
plt.title('UE WITH SERVICE AFTER BS LOSS WITH UABS IMPLEMENTATION', fontsize='18')
plt.grid(True)

print(service_ratio(itt['best_specimen']['specimen'].chromosomes,exp_pp.unserviced[1],200000
))
-----------------------------------------------

fig1 = plt.figure(1)

col = ['-*b','-og','-^r','-vc','-Dm','-sy','-Pk','-pb','-Xg']
lab = ['UBPP','UBPN','UBPU','UBNP','UBNN','UBNU','UBUP','UBUN','UBUU']

for j,k in enumerate((exp_pp,exp_pn,exp_pu,exp_np,exp_nn,exp_nu,exp_up,exp_un,exp_uu)):
  prt = []
  for cap in k.capacities:
     [a,b,c] = plt.hist([x[1] for x in cap],2)
     prt.append((a[1])*100/(a[0]+a[1]))
  fig2 = plt.figure(2)
  plt.plot([x for x in range(10,100,10)],[100-x for x in prt[::-1]],col[j],markersize=9, label=lab[j])
  fig1 = plt.figure(1)

fig2 = plt.figure(2)
plt.legend()
plt.xlabel('BS loss [%]', fontsize='18')
```

```
plt.ylabel('UE serviced after UABS Implementation [%]', fontsize='18')
plt.title('SERVICE DIFFERENCE FOR UE BETWEEN INITIAL LOSS AND AFTER UABS
IMPLEMENTATION', fontsize='18')
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.grid(True)
fig2.show()

print(service_ratio(exp_pp.results[5]['best_specimen']['specimen'].chromosomes,exp_pp.unser
viced[1],200000))
print([list(filter(lambda x: x==90,[x['loss'] for x in k.results])) for k in (exp_pp, exp_pn, exp_pu)])

-----------------------------------------------

def avg_distance(chromosome_set):
  '''
  The avg distance of all the paths for a given child (set of chromosomes)

  Parameters
  ----------
  chromosome_set : [chromosome]
     A set of chromosomes

  Returns
  -------
  (float)
     The sum of all the distances of all paths (chromosomes) divided by the number of
chromosomes
  '''
  import numpy as np

  suma=0
  for path in chromosome_set.copy():
     aa = [point for point in path][:-1]
     bb = [point for point in path][1:]
     for point_a, point_b in zip(aa,bb):
        suma += distance(point_a,point_b)
     suma += distance(aa[0],bb[-1])
  return(suma/len(chromosome_set))

plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
       [x for x in exp_pp.results]))]) for k in range(10,100,10)], sigma=1),'-*b',markersize=12,
label='UBPP')
plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
```

```
        [x for x in exp_pn.results]))]) for k in range(10,100,10)], sigma=1),'-og',markersize=12,
label='UBPN')
plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_pu.results]))]) for k in range(10,100,10)], sigma=1),'-^r',markersize=12,
label='UBPU')

plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_np.results]))]) for k in range(10,100,10)], sigma=1),'-vc',markersize=12,
label='UBNP')
plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_nn.results]))]) for k in range(10,100,10)], sigma=1),'-Dm',markersize=12,
label='UBNN')
plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_nu.results]))]) for k in range(10,100,10)], sigma=1),'-sy',markersize=12,
label='UBNU')

plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_up.results]))]) for k in range(10,100,10)], sigma=1),'-Pk',markersize=12,
label='UBUP')
plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_un.results]))]) for k in range(10,100,10)], sigma=1),'-pb',markersize=12,
label='UBUN')
plt.plot([i for i in
range(10,100,10)],gaussian_filter1d([min([avg_distance(y['best_specimen']['specimen'].chromos
omes)*60/50 for y in list(filter(lambda x: x['loss']==k,
        [x for x in exp_uu.results]))]) for k in range(10,100,10)], sigma=1),'-Xg',markersize=12,
label='UBUU')

plt.legend()
plt.setp(plt.gca().get_legend().get_texts(), fontsize='20')
plt.xlabel('BS LOSS', fontsize='18')
plt.ylabel('TIME [minutes]', fontsize='18')
plt.title('AVERAGE ROUND-TRIP TIME NEEDED PER BS LOSS', fontsize='18')
```