Self-organization for 5G and Beyond Mobile Networks using Reinforcement Learning

Paulo Henrique Valente Klaine

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Engineering
College of Science and Engineering
University of Glasgow

July 2019

Abstract

The next generations of mobile networks 5G and beyond, must overcome current networks limitations as well as improve network performance. Some of the requirements envisioned for future mobile networks are: addressing the massive growth required in coverage, capacity and traffic; providing better quality of service and experience to end users; supporting ultra high data rates and reliability; ensuring latency as low as one millisecond, among others. Thus, in order for future networks to enable all of these stringent requirements, a promising concept has emerged, self organising networks (SONs). SONs consist of making mobile networks more adaptive and autonomous and are divided in three main branches, depending on their use-cases, namely: self-configuration, self-optimisation, and self-healing. SON is a very promising and broad concept, and in order to enable it, more intelligence needs to be embedded in the mobile network. As such, one possible solution is the utilisation of machine learning (ML) algorithms. ML has many branches, such as supervised, unsupervised and Reinforcement Learning (RL), and all can be used in different SON use-cases.

The objectives of this thesis are to explore different RL techniques in the context of SONs, more specifically in self-optimization use-cases. First, the use-case of user-cell association in future heterogeneous networks is analysed and optimised. This scenario considers not only Radio Access Network (RAN) constraints, but also in terms of the backhaul. Based on this, a distributed solution utilizing RL is proposed and compared with other state-of-the-art methods. Results show that the proposed RL algorithm outperforms current ones and is able to achieve better user satisfaction, while minimizing the number of users in outage. Another objective of this thesis is the evaluation of Unmanned Aerial vehicles (UAVs) to optimize cellular networks. It is envisioned that UAVs can be utilized in different SON use-cases and integrated with RL algorithms to determine their optimal 3D positions in space according to network constraints. As such, two different mobile network scenarios are analysed, one emergency and a pop-up network. The emergency scenario considers that a major natural disaster destroyed most of the ground network infrastructure and the goal is to provide

coverage to the highest number of users possible using UAVs as access points. The second scenario simulates an event happening in a city and, because of the ground network congestion, network capacity needs to be enhanced by the deployment of aerial base stations. For both scenarios different types of RL algorithms are considered and their complexity and convergence are analysed. In both cases it is shown that UAVs coupled with RL are capable of solving network issues in an efficient and quick manner. Thus, due to its ability to learn from interaction with an environment and from previous experience, without knowing the dynamics of the environment, or relying on previously collected data, RL is considered as a promising solution to enable SON.

University of Glasgow

College of Science & Engineering

Statement of Originality

Name: Paulo Henrique Valente Klaine

Registration Number:

I certify that the thesis presented here for examination for a PhD degree of the University of Glasgow is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it) and that the thesis has not been edited by a third party beyond what is permitted by the University's PGR Code of Practice.

The copyright of this thesis rests with the author. No quotation from it is permitted without full acknowledgement.

I declare that the thesis does not include work forming part of a thesis presented successfully for another degree.

I declare that this thesis has been produced in accordance with the University of Glasgow's Code of Good Practice in Research.

I acknowledge that if any issues are raised regarding good research practice based on review of the thesis, the examination may be postponed pending the outcome of any investigation of the issues.

Signature: .....

Date: 12/07/2019

List of Publications

Journals

1. Yao Sun, Gang Feng, Lei Zhang, **Paulo V. Klaine**, Muhammad A. Imran, and Ying-Chang Liang. Intelligent Handoff for Radio Access Network Slicing by Exploiting Distributed Learning. Under review.

2. **Paulo V. Klaine**, Shufan Yang, Richard D. Souza, João P. B. Nadas, and Muhammad A. Imran. Positioning of multiple aerial base stations in pop-up networks using reinforcement learning. Under review.

3. Fauzun Asuhaimi, Shengrong Bu, **Paulo V. Klaine**, and Muhammad A. Imran. Channel Access and Power Control for Energy-Efficient Delay-Aware Heterogeneous Cellular Networks in Smart Grid Communications using Deep Reinforcement Learning. Minor Revisions (Submitted to IEEE Access).

4. **Paulo V. Klaine**, Mona Jaber, Richard D. Souza, and Muhammad A. Imran. Backhaul aware user-specific cell association using Q-learning. To appear in IEEE Transactions on Wireless Communications, 2019.

5. **Paulo V. Klaine**, João P. B. Nadas, Richard D. Souza, and Muhammad A. Imran. Distributed drone base station positioning for emergency cellular networks using reinforcement learning. Cognitive Computation, page 790–804, May 2018.

6. **Paulo V. Klaine**, Muhammad A. Imran, Oluwakayode Onireti, and Richard D. Souza. A survey of Machine Learning techniques applied to self-organizing cellular networks. IEEE Communications Surveys Tutorials, 19(4):2392–2431, July 2017.

Book Chapters

1. João Nadas, **Paulo Klaine**, Richard Demo Souza, Rafaela Parisotto, and Muhammad Imran. Intelligent Positioning of UAVs for Cellular Networks.

To appear in book titled: Enabling 5G Communication System to Support Vertical Industries, published by Wiley (July 2019).

2. Muhammad Ali Imran, Ana Flávia dos Reis, Glauber Brante, **Paulo Valente Klaine**, and Richard Demo Souza. Machine Learning in Energy Efficiency Optimisation. To appear in book titled: Machine Learning for Future Wireless Communications, published by Wiley, 2019.

3. **Paulo Valente Klaine**, Oluwakayode Onireti, Richard Demo Souza, and Muhammad Ali Imran. Next-Generation Wireless Networks Meet Advanced Machine Learning Applications, chapter The Role and Applications of Machine Learning in Future Self Organizing Cellular Networks, pages 1–23. IGI Global, 2019.

4. Rami Ghannam, **Paulo Valente Klaine**, and Muhammad Imran. Solar Photovoltaic Power Plants: Advanced Control and Optimization Techniques, chapter Artificial Intelligence for Photovoltaic Systems. Springer Nature, 2018.

Conference Proceedings

1. Rafaela de Paula Parisotto, **Paulo Valente Klaine**, João Pedro Battistella Nadas, Richard Demo Souza, Glauber Brante, Muhammad Ali Imran. Drone Base Station Positioning and Power Allocation using Reinforcement Learning. To appear in Proc. IEEE 16th International Symposium on Wireless Communications Systems (ISWCS) 2019.

2. Yusuf Sambo, **Paulo Valente Klaine**, João Pedro Battistella Nadas, and Muhammad Ali Imran. Energy minimization UAV trajectory design for delay-tolerant emergency communication. To appear in Proc. IEEE ICC 2019 Workshop - Intelligent Wireless Emergency Communications Networks: Theory and Applications.

3. João Nadas, **Paulo Klaine**, Lei Zhang, Guodong Zhao, Muhammad Imran, and Richard Souza. Performance Analysis of Early-HARQ for Finite Block-Length Packet Transmission. To appear in Proc. IEEE International Conference on Industrial Cyber-Physical Systems, IEEE ICPS 2019.

4. Yao Sun, Gang Feng, Lei Zhang, **Paulo Valente Klaine**, Muhammad Ali Imran, and Ying-Chang Liang. Distributed learning based handover mechanism for radio access network slicing with data sharing. In Proc. 53rd IEEE International Conference on Communications (ICC), pages 1–6, 2019.

5. M. Ozturk, **P. V. Klaine**, and M. A. Imran. Introducing a novel minimum accuracy concept for predictive mobility management schemes. In 2018 IEEE International Conference on Communications Workshops (ICC Workshops): 5G Ultra Dense Networks (5G-UDN), 2018.

6. M. Ozturk, **P. V. Klaine**, and M. Ali Imran. Improvement on the performance of predictive handover management by setting a threshold. In Proc. IEEE 86th Vehicular Technology Conf. (VTC-Fall), pages 1–5, Sep 2017.

7. M. Ozturk, **P. V. Klaine**, and M. A. Imran. 3D transition matrix solution for a path dependency problem of Markov chains-based prediction in cellular networks. In Proc. IEEE 86th Vehicular Technology Conf. (VTC-Fall), pages 1–5, Sep 2017.

Contents

List of Tables

List of Figures

List of Acronyms

2G	Second Generation
3D	Three Dimension
3G	Third Generation
3GPP	3rd Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
ABS	Almost Blank Subframes
AFR	Adaptive Frequency Reuse
AIW-PSO	Adaptive Inertia Weight Particle Swarm Optimization
AWGN	Additive White Gaussian Noise
BPNN	Back Propagation Neural Network
BS	Base Station
CAC	Call Admission Control
CAPEX	CAPital EXpenditure
CDMA	Code Division Multiple Access
CF	Collaborative Filtering
CREO	Cell Range Extension Offset
CSI	Channel State Information
D2D	Device to Device
DE	Differential Evolution
DP	Dynamic Programming
DT	Decision Trees
ECN	Emergency Communication Networks
EE	Energy Efficiency
EICIC	Enhanced Inter-Cell Interference Coordination
EIRP	Equivalent Isotropically Radiated Power
ESM	Energy Saving Mechanism
FFR	Fraction Frequency Reuse
FIS	Fuzzy Inference System
FLC	Fuzzy Logic Controller

FPC	Fractional Power Control
FQL	Fuzzy Q-Learning
GA	Genetic Algorithm
GD	Gradient Descent
GoS	Grade of Service
GPS	Global Positioning System
HHO	Horizontal HandOver
HMM	Hidden Markov Model
HO	HandOver
ICIC	Inter-Cell Interference Coordination
IoT	Internet of Things
IP	Internet Protocol
ITU-R	International Telecommunication Union - Radio
KPI	Key Performance Indicator
LAP	Low Altitude Platforms
LoS	Line of Sight
LTE	Long Term Evolution
LTE-A	Long Term Evolution - Advanced
M2M	Machine to Machine
MC	Markov Chains
MCM	Monte Carlo Methods
MDP	Markov Decision Process
MDT	Minimization of Drive Tests
MIMO	Multiple Input Muplitple Output
ML	Machine Learning
MLB	Mobility Load Balancing
cMTC	Critical Machine-Type Communications
MOO	Multi-Objective Optimisation
MRO	Mobility Robustness Optimization
MSE	Mean Squared Error
NCL	Neighbour Cell List
NFV	Network Function Virtualisation
NGMN	Next Generation of Mobile Networks
NLoS	Non-Line of Sight
NN	Neural Network
NNS	Nearest Neighbour Search
NP-hard	Non-deterministic Polynomial-time hard
OFDMA	Orthogonal Frequency Division Multiple Access

OPEX	OPerational EXpenditure
PSO	Particle Swarm Optimisation
QoE	Quality of Experience
QoS	Quality of Service
RAN	Radio Access Network
RB	Resource Block
RL	Reinforcement Learning
RLF	Radio Link Failure
RNC	Radio Network Controllers
RSRP	Reference Signal Received Power
SARSA	State-Action-Reward-State-Action
SC	Small Cell
SINR	Signal to Interference Plus Noise Ratio
SOM	Self Organising Map
SON	Self Organising Networks
SVM	Support Vector Machine
TD-Learning	Temporal-Difference Learning
TL	Transfer Learning
TTT	Time to Trigger
UAV	Unmanned Aerial Vehicle
UAV-BS	Unmanned Aerial Vehicle Base Station
UE	User Equipment
UHD	Ultra High Definition
V2X	Vehicle to Anything
VFA	Value Function Approximation
VHO	Vertical Hand Over
VR	Virtual Reality

Acknowledgements

First and foremost, I would like to express my deepest gratitude for my supervisor, Prof. Muhammad Ali Imran for supporting me during this three year journey that was the PhD. I would also like to thank Prof. Imran for providing funding during my PhD, as well as for all the discussions and valuable feedback given for all publications. Lastly, thank you for always keeping me motivated, being patient and sharing your knowledge so that I could be a better researcher.

Secondly, I would like to thank Dr. Richard Demo Souza, from the Federal University of Santa Catarina, Brazil, who has been a collaborator and also a guide during my PhD. Although not being an official supervisor, Dr. Souza acted as one and provided more than enough guidance, knowledge and feedback during all 3 years of the PhD, and for that I am extremely grateful.

I would also like to thanks Dr. Shufan Yang, for sharing her knowledge on machine learning, more specifically in reinforcement learning. Thank you for all the guidance, help and for sharing your knowledge in this field, which motivated me to study and learn the algorithms in this field.

I am also very thankful for my second supervisor, Dr. Wasim Ahmad, who gave some very good suggestions and key insights during my work and also in this thesis. I am also grateful for Dr. Qammer Abbasi, for his discussions and feedback on this thesis.

I would also like to acknowledge the support and help of all my friends during the PhD, without them this journey would have been much harder and arduous. Thank you all for the discussions in and out the university. I would also like to thank my girlfriend, Yanmei Cao, for supporting me, and keeping me motivated during these past 7 months.

Last but not least, I would like to thank my family for all the support throughout this journey, in special my parents, Marcia Klaine and Geraldo Klaine, and my sister, Anna Klaine.

Chapter 1

Introduction

In the last 10 years, mobile networks have become an essential part of our lives, due to a broad range of applications and services that have recently become available. For instance, people are able to do business on the go by performing teleconferences whenever and wherever needed, watch their favourite videos and listen to their favourite music on the fly, talk to distant relatives, stream audio/video whenever a special event happen, instantly upload photos or videos about their daily lives in social media, and many more [8, 9, 11].

Due to the increasing popularity of mobile services, its traffic is expected to grow around 10 thousand times, and the number of devices connected to the network is expected to be around fifty billion by the next few years [12–14]. Because of the exponential growth that is expected in both connectivity and density of traffic, primarily due to the advances in the Internet of things (IoT), machine-to-machine (M2M) communications, cloud computing and many other technologies, the fifth generation (5G) of mobile communications and beyond will need to push the network performance to a next level. Furthermore, 5G will also have to address current limitations of long term evolution (LTE) and LTE-advanced (LTE-A), such as latency, capacity and reliability. As such, some of the requirements that are recurrent in state-of-the-art literature for 5G networks are [8, 9, 15–17]:

- Address the growth required in coverage capacity and traffic;
- Provide better quality of service (QoS) and quality of experience (QoE) to users;
- Support the coexistence of different radio access network (RAN) technologies;
- Support a wide range of applications;

- Provide peak data rates of over 10 Gbps and at least 100 Mbps at the cell edge;
- Support radio latency lower than one millisecond;
- Support ultra high reliability;
- Provide improved security and privacy;
- Provide more flexibility and intelligence in the network;
- Reduction of capital and operational expenditures (CAPEX and OPEX);
- Provide higher network energy efficiency (EE).

As it can be seen, these requirements are very stringent. Hence, in order to meet them, new technologies have to be deployed in all layers of the 5G network. As a result, several breakthroughs have been and are being discussed in the literature for the past couple of years, the most common ones are: massive MIMO (multiple-input multiple-output), millimeter-waves (mm-waves), new physical layer waveforms, network function virtualization (NFV), control and data plane separation, network densification - deployment of several small cells (SCs) - and implementation of self organizing networks (SON) functions [8, 16].

Although all of these breakthroughs are important and often considered a necessity for future mobile networks, the concept of network densification is the one that requires heavier changes in the network and possibly a change in paradigm in terms of how network solutions are provided [18]. In addition, the deployment of several SCs would most likely address the current limitations of coverage, capacity and traffic demand, while also providing higher data rates and lower latency to end users [8, 16]. However, while network densification will result in all these benefits, it will also generate several new problems to the operators in terms of coordination, configuration and management of the network. The dense deployment of several SCs, will result in an increase in the number of mobile nodes that will need to be managed by mobile operators. Furthermore, these types of cells will also collect an immense amount of data in order to monitor network performance, maintain network stability and provide better services. This will result in an increasingly complex task to configure and maintain the network in an operable state if current techniques of network deployment, operation and management are applied [8, 19].

One possible way of solving these issues is by deploying more intelligence in the network, through SON. The main objectives of SON can be defined as to provide intelligence to mobile networks in order to make the work of operators easier, as

well as provide network resilience, reduce the overall network complexity, CAPEX and OPEX, and simplify network coordination, configuration, optimization and healing [8, 9, 11, 20].

1.1 Self Organising Networks (SON)

SONs, in contrast to previous generation of mobile networks, envision that future networks will be coupled with intelligent algorithms in order to keep the network performance near their optimal point. This is done through the collection and analysis of data, and will enable future mobile networks to be much more proactive and flexible, as the networks will be able to learn by themselves, without human's intervention.

More specifically, SON can be defined as an adaptive and autonomous network that is also scalable, stable and agile enough to maintain its desired objectives [11]. Hence, these networks are not only able to independently decide when and how certain actions will be triggered, based on their continuous interaction with the environment, but are also able to learn and improve their performance based on previous actions taken by the system. Furthermore, the concept of SON in mobile networks can also be divided into three main categories, mainly: self-configuration, self-optimization and self-healing; and together they are commonly denoted as *self-x* functions [9, 11].

1.1.1 Self-configuration

Self-configuration can be defined as the ability to execute all the configuration procedures to make the network operable autonomously, that is, with no human intervention [8]. Examples of configuration parameters that can be configured autonomously can be individual base stations (BSs) Internet protocol (IP) address, neighbour cell list (NCL), radio and cell parameters, or parameters that will be applied to the whole network, such as policies. Self-configuration is mainly triggered whenever a new base station is deployed in the system, but it can also be activated if there is a change in the network, as a BS failure or change of service or network policies.

1.1.2 Self-optimization

After the system has been configured, the self-optimization functions are triggered. Self-optimization functions can be defined as procedures that continuously optimize the BSs and network parameters in order to guarantee a near op-

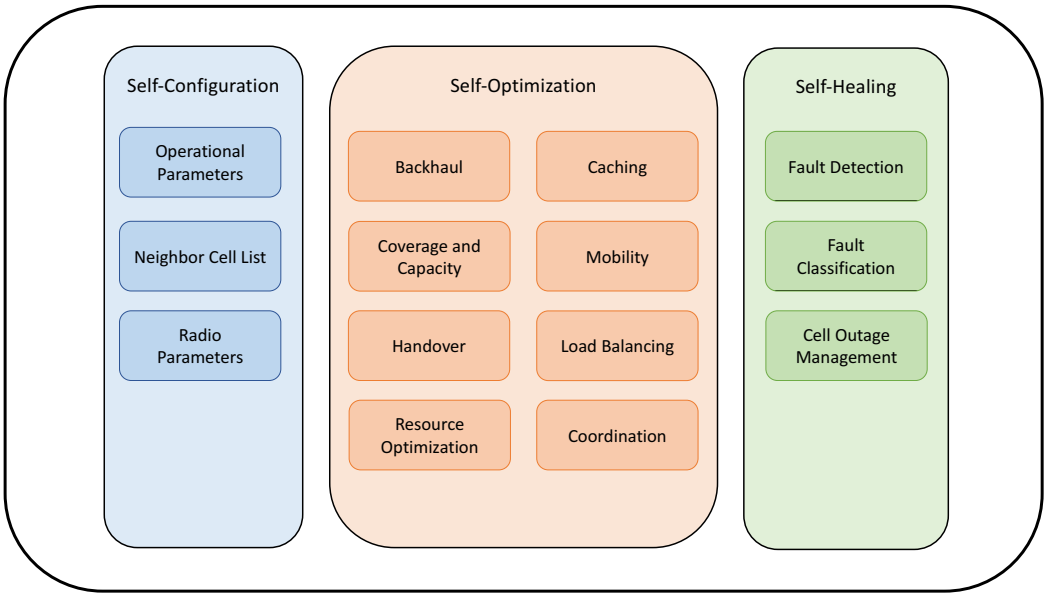


Figure 1.1: Major use cases of each SON function: self-configuration, self-optimization and self-healing [8].

timal performance. Self-optimization can occur in terms of backhaul parameters, caching, coverage and capacity, antenna parameters, interference management, user mobility, hand over (HO) parameters, load balancing, resource optimization, call admission control (CAC), EE, and coordination of SON functions [8]. By monitoring the system continuously, and using reported measurements to gather information, self-optimization functions can ensure that the network objectives are maintained and that its overall performance is kept near optimum.

1.1.3 Self-healing

In parallel to self-optimization, the function of self-healing can also be triggered. Since no system is perfect, faults and failures can occur unexpectedly and it is no different with mobile systems. Whenever a fault or failure occurs, for whatever reason (e.g., software or hardware malfunction), self-healing functions are activated. Their objective is to continuously monitor the system in order to ensure a fast and seamless recovery. Self-healing functions should be able not only to detect failure events but also to diagnose the failure (i.e., determine why it happened) and also trigger the appropriate compensation mechanisms, so that the network can return to function properly. Self-healing in mobile systems can occur in terms of network troubleshooting (fault detection), fault classification, and cell outage management [11, 21–23].

1.1.4 SON Use-cases

Each SON function can also be divided into sub-sections, commonly known as use-cases. Figure 1.1 shows an outline of the most common use cases of each SON task. As it can be seen from Figure 1.1, future mobile networks are expected to address several different use-cases and provide many solutions in domains that either do not exist today or are beginning to emerge [8]. For example, in current networks whenever a new BS is deployed in the network, hundreds of parameters need to be manually configured by expert engineers in order for it to become operable. As such, just the simple task of adding a new BS in a network requires a lot of expertise and manual work, which represents an increased cost. In addition, most of these parameters can also be sub-optimally configured, as the radio parameters of a BS can also affect the total network coverage or its overall topology. One can think of as an example, if the operator configures the power of the new BS too high, this will have an impact on nearby BSs, as their coverage area will be reduced and interference will increase. Thus, a simple miscalculation or change in network conditions can be disastrous [9].

This simple example also highlights that a static configuration of network parameters is not necessarily optimal, as the network is constantly changing, thus, it is also necessary to constantly evaluate and optimize the parameters previously configured. However, as it is with the configuration process, the optimization process of a network, nowadays, also relies on manual intervention. Furthermore, the constant monitor of performance indicators of the network is still required in order to determine which parameters of a cell need to be updated and by how much [8, 20]. Lastly, healing of mobile networks also depends on manual interventions. By constantly monitoring network alarms and network performance, operators are able to detect whenever network failures occur and send expert engineers to the field in order to address these problems. However, this process is not optimal, as it requires a lot of expertise, and constant analysis as well as monitoring of the network in order to react to a failure, which can be extremely ineffective and costly. Moreover, certain types of failures also take a long time to be detected, such as the problem of sleeping cells, for example, which can cause severe service disruption in an area for days or weeks before the operator has become aware of the problem, generating a big loss in revenue for operators [8].

In order to address these use-cases, several research groups are implementing intelligent solutions and also standardizing methods to improve future networks, as it can be seen from The 3rd Generation Partnership Project (3GPP), Next Generation Mobile Networks (NGMN) Alliance, mobile operators and many other research initiatives. Current state-of-the-art algorithms to tackle the aforemen-

tioned use-cases go all the way from basic control loops and threshold comparisons to more complex machine learning (ML) and data mining techniques [24]. However, as the field develops, there is a significant trend of implementing more robust and advanced techniques which would in turn solve more complex problems [25]. As such, by coupling mobile networks with intelligent algorithms, all of the aforementioned issues can be solved in a more efficient, optimal and cost-effective way.

1.2 Motivation

The next generations of mobile networks are under heavy pressure in order to address their expected requirements and also improve upon current network limitations. Such limitations, for example, consist of the rudimentary methods utilized today in the configuration, optimization and healing of mobile networks, as these methods lack the adaptability and flexibility required to become feasible solutions for 5G and beyond networks [8,9]. In addition, despite mobile operators collecting a huge amount of data daily from the network, it has been shown that most of the collected data is not utilized or even discarded, limiting the potential of SON and leading to sub-optimal solutions [19]. As such, many of the solutions implemented today require human intervention, such as expert engineers to analyse data and adjust system parameters manually in order to optimize or configure the network. Some other solutions also require expert personnel on site in order to fix certain problems, when detected, wasting a substantial amount of time and resources. All these solutions are extremely ineffective and costly to mobile operators [20]. Thus, in order to leverage all the information already collected by operators and provide the network with adaptable and flexible solutions, it is clear that a new paradigm is needed, such as SON [8].

1.2.1 Why SON is needed?

In the last decade, with the advance of the technology industry and the Internet, the ability for people to connect to each other and share their experiences has become an essential part of our lives [11]. Furthermore, in the last couple of years we have seen a plethora of new applications and services available through the Internet, services that a couple of years ago were unimaginable. For example, we can now talk to distant relatives instantly, share our daily lives and experiences to all our friends in social media, order food and cars with the touch of a button, as well as watch our favourite videos and shows anywhere and at anytime we

want [8, 9, 11].

However, the future of mobile communications will see a great change, due to other technological advances, such as the IoT, M2M, smart cars and cities, virtual reality (VR) and ultra high definition (UHD) videos, cloud computing and many more. As such, several other applications that are unimaginable today are bound to happen, and for that to be possible, a drastic change in mobile network structure is needed [8]. This change will mainly occur because, in the future, it is expected that many more use-cases will be enabled by future mobile networks, demanding very stringent requirements from mobile networks [8, 20]. For example, in the case of VR applications, extremely high bandwidth with low latency is required in order to guarantee the desired QoS to end-users. Another example is the use-case of critical machine-type communications (cMTC), in which a whole industry is connected via wireless networks. In this case, delay and reliability are extremely important so that machines can work properly and keep the production line going [26]. Another use-case that can be enabled in the future by mobile networks is the case of vehicle-to-anything (V2X) communications, in which applications can demand extremely low latencies and ultra high reliability in order to avoid car accidents, as well as a large capacity in order to provide infotainment services to users inside the car [26].

However, while these applications and use-cases are only limited by the human imagination, mobile networks have their own limitations, which are restricted by physical restrictions as well as financial constraints from network operators, as higher capacity and QoS comes at the expense of higher costs [20]. Furthermore, since end-users are not willing to pay higher bills for improved mobile services, minimizing network costs, while still providing better QoS and capacity to end-users is crucial in order to make future networks viable [8, 20]. As a result, this trade-off between providing improved services in a cost effective way has triggered research to add more intelligence to mobile networks, which is mainly motivated by the following factors [8, 20]:

1. Current mobile networks lack the adaptability and flexibility necessary in order to adapt to the variations and fluctuations in the network. Due to the natural unpredictability of the wireless channel, dynamic requirements of users and their mobility, current mobile networks suffer either from underutilization (for example during night time or non-peak hours) or overutilization of network resources (such as in hot-spot areas, or whenever events happen in a city), resulting in a low resource efficiency and congestion, as well as poor QoS and QoE to end-users, respectively;

2. 5G networks are envisioned to have a network densification process, in which the network will be populated by a large number of SCs, that can be deployed not only by the network operators but also by end-users. As such, this will create a tremendous burden for operators, as they will have to manage, configure and optimize an astounding number of SC parameters in order to guarantee the correct operation of the network. As this will not be humanly possible, more optimized and intelligent approaches are needed;

3. Another consequence of the network densification process is that future mobile networks are expected to increase exponentially in size, resulting in a much more complex and challenging network to be managed. As such, given the huge scale of these systems, the classic approach of manual optimization, periodic drive tests and inspections will be extremely inefficient. Thus, more robust and intelligent solutions are preferred;

4. Lastly, SON can also significantly reduce the operational costs of network, as it can replace the periodic tests and field analysis required by expert engineers, as well as the analysis and monitoring of the data generated by the network daily.

As it can be seen, SONs are the only cost effective way to achieve near optimal performance in future mobile networks, as they are able to automatically configure, optimize, maintain, troubleshoot and recover the network, eliminating the humans in the loop [8, 11].

In addition to these factors, another motivating aspect is the huge amount of data that is generated and collected by network operators on a daily basis, due to the large amount of sensors and monitoring required [8, 19, 20]. However, as shown in [19], despite mobile networks generating a huge amount of data everyday in terms of network measurements, control and management interactions, drive tests and subscriber data, most of this information is not currently being used. In addition, because storing such a great amount of data is costly, network operators normally discard most of the data generated after its usage, leading to potential resource wastage, as this data could be leveraged by intelligent algorithms to learn network patterns and behaviour in order to provide better solutions [19].

As a result, current state-of-the-art approaches for network operators consist of collecting and storing only a small amount of information, such as management information (such as network performance or failures) and customer relationship information (such as complaints about bad service or churn information) [19]. In addition, most of the current solutions in mobile networks are done manually, such as the configuration of a newly deployed BS, periodic drive tests in order to check

network conditions, intervention of engineers and field personnel in order to fix network faults, etc. [8, 11, 19]. On the other hand, the small amount of solutions that are performed autonomously rely on simple and low-complexity solutions, such as comparing to a threshold, basic feedback controllers, or heuristic and search algorithms [8, 19].

However, as it can be seen, these solutions are quite costly and inefficient for network operators. In addition, they also lead to sub-optimal network configurations as the network is constantly changing, and current networks do not have the capability to dynamically adapt themselves. Furthermore, with the advent of network densification, the amount of parameters that will need to be configured will be tremendous and the amount of data generated by future networks will be even larger, rendering current manual solutions impracticable [8, 11, 19]. As such, more robust and data oriented solutions, that involve algorithms that can analyze and find relationships and intrinsic patterns in data, are required. By combining future mobile networks with intelligent solutions, the next generation of mobile networks will be able to operate at its full potential, with minimum resource wastage, while also reducing its costs [8].

1.2.2 Why ML is needed?

Recent advances in technology, such as the miniaturization of electronic devices, the ever increasing computing power of machines and the instant connectivity provided by the Internet anywhere and at anytime in the world, has enabled companies to capture trillions of bytes of information everyday, in what is known as *Big Data* [27–29]. Such information comes from the billions of sensors connected to everyday objects, such as our mobile phones, automobiles, home appliances, and computers that are able to sense, create and communicate data [27]. As such, due to the massive volume of data generated and collected every second, it was only natural that companies would try to explore that to their benefit using data analytics, in order to create better and more profitable solutions that would also please their customers [27, 29].

ML algorithms rely on collecting and analysing data in order to find intrinsic patterns and relationships between them and produce a model that can relate the input to the output, instead of trying to develop a complex and complete model of the system [8]. As a result, these algorithms are able to learn, reason and make decisions without human intervention. In addition, due to trends such as the Moore's Law in computing, its equivalent in digital storage and recent advances in electronics and cloud computing, the ability to store, process and analyse data, also contributed to the rise in popularity of ML algorithms in recent years

[8, 27, 30]. In addition, another clear advantage of ML is that these algorithms are able to generalise [8, 31, 32]. For example, considering the task of mobile network optimization, if a model had to be developed for every possible situation of the network, with users in all possible positions and all BSs with different power, interference and load levels, it would be impossible. As such, solutions that analyse data and are able to create a model based on their observations are much more feasible. This occurs because ML solutions are able to learn from data and make predictions if new unseen data is fed into the model, being much more general than analytical approaches, as they do not require the entire model to be trained again or rebuilt from zero [31, 32].

Furthermore, another key advantage of ML algorithms is when dealing with complex tasks. Similarly to the case above, in which for some applications it is impossible to create an analytical model for every possible solution, in very complex domains, traditional approaches would also not work [8, 9]. For example, considering the task of teaching a self-driving car how to drive. The car has so many sensors, inputs, images from different cameras and also lots of possible actions, such as accelerating, braking, turning the wheel, changing gears, etc.. As it can be seen from this example, when tasks are extremely complicated and have a lot of variables and parameters, traditional analytical approaches or controller design solutions are not very suitable, as the solutions required to solve these problems would be too complex and costly [8, 9]. As such, in those cases ML solutions also excel at, as they are able to learn from the great amount of data gathered and generated from these complex applications and determine the best action.

In summary, the main advantages of ML over traditional and analytical methods are:

- The volume and speed that data is generated today has seen a massive increase in the last few years, as such, the process of collecting, storing and analysing data has become infeasible at a human scale. If conventional data analysis techniques are applied to big data, the process of analysing and generating adequate responses would possibly be too inefficient, slow and sub-optimal, due to the sheer complexity and volume of data. As such, ML algorithms can enable a more efficient data analysis and generate near optimal results.

- ML algorithms are capable of analysing and processing a huge amount of previous stored historical data and learn from it. By analysing previous data, these algorithms are capable of finding hidden patterns and correla-

tions, generating a model that best represents all the observed data points. As a result, these algorithms are able to produce very robust solutions that are capable of generating accurate responses and future predictions about a model.

- Instead of relying on a fixed model, or having to develop a model for every possible new situation, ML algorithms enable generalization, or in other words, its models can be optimized online, constantly being improved and enhanced in order to learn new trends or drifts in data.

- ML is also able to deal with more complex problems in a much easier way. Instead of creating complex and intricate models of a particular problem, ML solutions rely on data analysis in order to fit models to the observed data set.

- ML algorithms, and more recently deep learning, have shown comparable human performance in certain tasks, such as image classification or playing certain games like chess, Go, backgammon, and video games [33–36]. As such, with the constant development of more robust and powerful computers and algorithms, the possibilities of what these intelligent algorithms can do are practically unimaginable.

1.2.3 Why Reinforcement Learning (RL)?

Based on the aforementioned issues, it is clear that analytical approaches are not ideal to tackle future mobile network use-cases. As such, ML solutions have to be explored in order to deal with the massive amount of data generated by mobile networks, as well as its increasing complexity [8].

Although ML has several branches, this thesis focuses on the application of RL in SON. The idea of learning by experience or by interaction with the environment is probably the first thought that comes to mind when thinking about the nature of learning [10]. For example, when a child is learning how to walk, this task has no explicit teacher, but actually there is a direct sensorimotor connection between the agent and its environment [10]. By exercising this connection, a huge amount of information is produced about the consequences of different actions, cause and effect and what it needs to be done in order to achieve certain goals. From this simple example, it can be seen that the interaction between agents and an environment are a major source of knowledge, not only about the environment, but also about the agent itself. As such, learning from interaction is a fundamental idea that underlies nearly all theories of learning and intelligence [10].

RL algorithms try to explore this idea of interaction by learning via a goal-seeking, or trial and error, approach in order to find its solutions [10]. Because the agent is inserted in an environment and is experiencing it, the agent is able to interact with it, learn and adjust their behaviour based on past experiences, building intelligence over time. Furthermore, because of this trial and error approach, RL solutions also pose another advantage that they do not require to know a model of the environment (the RL class of Temporal Difference learning algorithms) in order to work (these solutions are said to be model-free) [10]. Because the agent is only concerned on learning what to do, or in other words, only mapping situations to actions in order to achieve its objective, RL algorithms do not require previous knowledge about the environment, as it can learn from its own experience [10].

Another key advantage of RL algorithms is their ability to explicitly consider the whole problem instead of dividing it into smaller sub-problems. This occurs because agents in RL have explicit goals that they must follow, given by the reward function. By doing so, they can interact and sense the environment, choosing actions that influence the way the environment responds to them [10]. As such, the key features of RL, which motivated its applications in SON use-cases, can be summarized as follows:

- RL algorithms based on temporal differences do not require a model of the environment or any previous knowledge about it, as they are able to learn from experience and interaction with the environment due to their inherit goal-seeking approach;
- This interaction between agent and environment enables RL agents to adjust their behaviour and learn from past experiences, instead of relying in previous examples and data provided by an external and knowledgeable supervisor;
- RL agents are programmed with a clear goal in mind (given by the reward function), as such they are able to tackle the whole problem instead of dividing it into smaller sub-problems. This makes RL algorithms particularly suitable for problems that include a long-term versus short-term reward trade-off, or optimisation, as is the case of mobile networks.

1.3 Objectives

As previously mentioned, this thesis focuses on the application of RL algorithms in order to tackle SON use-cases. More specifically, RL algorithms are utilized

in order to perform self-optimization of future mobile networks. In this realm, different RL solutions are tested in different scenarios and compared to other current state-of-the-art approaches. Albeit different, these scenarios share a common feature, which is the main issue that this thesis tackles, which is the end-to-end optimisation of network resources through RL algorithms.

However, before focusing on the application of RL algorithms in SON, the first objective of this thesis is to present a literature review of ML algorithms applied in SON. Since SON is a very broad domain, with its three main functions (self-configuration, self-optimization and self-healing), and the thesis only focuses on the area of self-optimization, an extensive literature review of this field is presented. In addition to a literature review in the area of SON, a brief background on ML covering the areas of supervised and unsupervised learning is presented. This is followed by a broad overview of RL, the focus of this thesis, in which the main algorithms are presented and advantages and drawbacks are discussed.

Regarding the application of RL in self-optimization use-cases, another objective of this thesis is in the context of the optimisation of user-cell association procedures, considering end-to-end connectivity. As such, in order to deliver better experiences to end-users and use network resources efficiently, a joint optimisation is proposed, in which radio access and backhaul resources are jointly optimised in order to deliver a better cell association. As it will be seen in the literature review section, the optimization of the backhaul is an important issue in future mobile networks, however, despite this fact, not many researchers have investigated this research topic [8]. As such, one of our objectives is to attempt to tackle this problem through the utilization of RL algorithms in order to improve the overall performance of the network.

Another emerging topic in the realm of self-optimization in SON is the utilization of unmanned aerial vehicles (UAVs), as movable BSs, in order to provide additional coverage or capacity for mobile networks. Due to their mobility and line-of-sight (LoS) communication capabilities, UAVs are envisioned to play a key role in future mobile networks, however one of the main issues that still remains unanswered is how to position multiple UAVs in a mobile network in order to maximize coverage. In this thesis an attempt to solve this issue by deploying intelligent UAVs coupled with RL algorithms and limited by network constraints is proposed. In order to evaluate the proposed solutions, two different simulation scenarios were evaluated. First, an emergency communication scenario was envisioned. This consists of an area in which part of the communication infrastructure was destroyed due to a natural disaster. As such, the deployment of UAVs is performed to provide the necessary coverage by optimising the 3D posi-

tion of multiple UAV BSs considering end-to-end user requirements and network constraints. In this case, UAVs need to be embedded with intelligent algorithms in order to be deployed in a quick and effective manner, to restore and provide network service as quickly as possible in order to potentially save human lives. The second scenario consisted of an event happening in a random part of a city, which causes network congestion due to the excessive amount of people concentrated in a specific area (also known as pop-up networks). As such, in this other case, UAVs are deployed in order to enhance network coverage and capacity to provide service to end-users, while avoiding or minimizing interference with the ground network. In addition, UAVs are also limited by their own resources and need to jointly optimise their positions considering different user requirements and aerial and ground network constraints. Based on these two UAV scenarios, different RL algorithms are tested and the proposed solutions are compared with other state-of-the-art techniques.

In summary, the objectives of the thesis can be outlined as follows:

- Explore the utilization of RL algorithms in different SON use-cases, with the focus on self-optimization in future mobile networks, considering scenarios in which end-to-end connectivity is required and network constraints are limiting;
- Provide a literature review on the application of ML techniques in the realm of self-optimization in SON, analysing previous works and classifying them according to their learning technique and use-case;
- Analyse the impact of RL solutions in a proposed backhaul optimization use-case considering not only RAN, but also backhaul constraints and users requirements in order to achieve a better user-cell association;
- Analyse the impact of different RL algorithms in the context of UAVs applied in mobile networks in two different SON use-cases, one considering an emergency situation and another consisting of a pop-up network, both considering UAVs limited by network constraints;
- Provide future trends and research directions as well as conclusions in the topic of RL and ML applied in SON.

1.4 Research Contributions

Based on the aforementioned objectives, this thesis focus on the application of RL algorithms in self-optimisation use-cases of mobile networks in which end-to-end

requirements and network constraints are considered. As such, three different optimisation scenarios are investigated and the application of different RL algorithms is considered, their performance is evaluated and compared to other state-of-the-art solutions. Based on that, the contributions of this thesis can be summarized as follows:

1. Provide an extensive literature review of ML algorithms applied in SON. This literature review encompasses the last 15 years of research performed in the area and each work is classified according to their learning type as well as the SON use-case it tackled. Furthermore, some suggestions and guidelines of the application of ML algorithms in SON use-cases are given, as well as potential future research directions in the area of SON. For details see [8].

2. Perform the optimization of user cell association considering backhaul constraints using a two-step Q-Learning algorithm. In this contribution the optimisation of parameters both from the network and end users is proposed, with the objective of enabling a user-specific cell association. This association depends on users' requirements and what each cell of the network had to offer (in terms of backhaul), users would be associated to the most fitting cell. Results show that the proposed approach performs better than conventional state-of-the-art solutions, or utilizing a Q-Learning solution to optimize only network side parameters.

3. Perform the optimization of coverage and capacity utilizing UAVs coupled with RL algorithms in an emergency communications scenario. In this case, it is envisioned that a ground network was destroyed and UAVs are deployed to provide service for stranded users as well as search and rescue teams. Based on this, a three dimension (3D) position optimization of multiple UAVs, based on network constraints and user requirements is developed utilizing Q-Learning. Results show that the proposed approach is robust, dynamic and agile, and that it is able to outperform other methods.

4. Similarly to the previous contribution, the next contribution also envisions the utilization of UAVs in order to optimize network coverage and capacity. However, this time, a pop-up network scenario was investigated, in which UAVs have to provide service for a large concentration of users that cannot be supported by the ground network. As such, the problem of 3D positioning multiple UAVs, constrained by user requirements, as well as interference from the ground network is considered. For this problem, two RL

techniques were proposed, one based on conventional RL, Q-Learning, and another one based on the concept of value function approximation (VFA). Results show that the two intelligent solutions are robust and are able to restore the network to its normal performance levels. In addition, results also show that the solution based on VFA is also able to outperform classic Q-Learning, highlighting the potential and advantages of VFA over traditional RL methods.

5. Lastly, this thesis finalises with some conclusions and future research directions in the realm of RL, as well as SON and ML applied in SON.

In addition to the main contributions, I have accomplished several other parallel tasks, which also culminated in other publications and contributions. For example, the works with Ozturk, et al. involved the self-optimization of mobile networks in terms of HO parameters and user mobility management (Conference Proceedings 5–7). These works were performed using another RL technique, Markov chains (MC), and attempted to reduce the HO cost of mobile networks by trying to predict to which BSs users would move to next. Furthermore, another work is (Conference Proceedings 4), in which RL (Q-Learning) is utilized in a scenario with network slicing in order to reduce HO costs of the network, while maintaining users QoS requirements. Another work involving Q-Learning is the one in (Conference Proceedings 1), in which an extension of the (Journal 5) is proposed and a joint optimisation of UAV position and transmit power is performed. Lastly, in (Conference Proceedings 2), UAVs are utilized in a scenario of emergency communication networks in order to provide delay tolerant communications. However, because this optimization was performed offline (as the location of temporary BSs was assumed to be fixed), a genetic algorithm (GA) approach was preferred instead of RL.

1.5 Thesis Outline

The remainder of this thesis is as follows. Chapter 2 starts by presenting an overview of current mobile networks and how 5G and future networks are expected to overcome present drawbacks. Then, an overview of current state-of-the-art ML techniques applied in SON use-cases is presented, with the focus on the application of ML in self-optimization. After that, an overview of ML is presented, focused on RL and its algorithms (that are used in the next chapters in the optimization of SON use-cases). Chapter 3 presents a use-case of self-optimization of the backhaul connection utilizing RL algorithms. This chapter starts with

a brief introduction and motivation for the problem, followed by a literature review on the topic. After that the system model is presented, followed by the proposed solution. Results are then assessed, comparing with other state-of-the-art approaches and conclusions are drawn.

Chapter 4 presents another use-case of self-optimization in SON, but this time utilizing intelligent aerial platforms (UAVs) in order to optimize the coverage and capacity of mobile networks in the context of emergency scenarios. This chapter starts by introducing the problem of coverage in the case when a natural disaster happens in a certain area. For the proposed scenario, a brief motivation is presented, followed by a recent state-of-the-art literature review. After that, the system model and proposed solutions are showcased, followed by their results. Similarly, Chapter 5 presents the utilization of intelligent UAVs in the context of pop-up networks, when high population density events, such as music concerts, open markets or sport events happen. A brief introduction to the problem is presented, followed by a literature review and a discussion on why analytical solutions are not practical for pop-up network scenarios. After that, a motivation is presented and the contributions of the proposed approaches based on RL are highlighted. Then, the system model, consisting of both the ground and aerial network is presented and the problem is formulated. The proposed solutions are then shown, results are showcased and the algorithms convergence properties analysed. Lastly, conclusions are drawn. Finally, Chapter 6 discusses future trends in RL and also the application of RL algorithms in SON, and also draws conclusions about the importance of learning, more specifically RL, in future mobile networks.

Chapter 2

Background and Literature Review

2.1 Overview in Current and Future Mobile Networks

As previously mentioned, current mobile networks rely heavily on human expertise in order to collect and interpret data to perform basic network functions, such as configuration, parameter optimisation or healing. This, by its turn, leads to inefficient and sub-optimal solutions that incur in a lot of costs for mobile operators, reducing revenue and limiting the network performance [8, 11]. As such, there must be a shift in paradigms in future networks towards a more autonomous and adaptable design. This should be performed in order to meet future network requirements and to deal with its increased complexity, as the network scales up, as well as keep on par with current technologies [8]. In the next few sections, a brief overview of current mobile networks is presented, followed by an outline of future technologies that will be a core part of future mobile networks, and which will need a shift in paradigms in order to be fully functional.

2.1.1 LTE and LTE-Advanced (4G)

Previous and current generations of mobile networks are mostly based on mathematical or statistical models derived from theory or from field measurements [37]. These models range all the way from operators analysing traffic patterns to determine where to deploy new BSs, analysing call data records to optimise different BSs, performing period drive tests to determine which BSs need to be fixed or even utilizing wireless channel models to calculate and determine parameters of the communication's link budget [8, 37]. Other examples include when operators

want to deploy a new network inside a building, expert engineers are sent in order to collect data about the scattering of the wireless signal, path loss, LoS, interference patterns, etc., and build a model of it. After a thorough analysis, the best location for one (or multiple) hot-spots is (are) chosen and the operator can go back to the site in order to perform its installation [8, 9].

As it can be seen by this example, mathematical models are at the heart of mobile communications, playing a significant role in all aspects, such as network design and configuration, network optimization and healing. However, this conventional approach has several drawbacks, such as [37]:

- Depending on the network complexity, or the complexity of the scenario being considered, it is really difficult to develop an accurate mathematical model. In addition, even if it were possible to create a mathematical model for every situation, there is also the trade-off between the accuracy of the model and its complexity, in which simplistic models cannot describe the observations well enough, while more accurate models need to be inherently more complex;

- The deployment and optimization of fixed network infrastructure might not be feasible for future use-cases of the future mobile networks, as current network designs lack the necessary adaptability and flexibility in order to handle dynamic changes in the environment. As such, the efficiency of current mobile networks are limited by their static design.

As it can be seen, these two issues can be very limiting for future mobile networks, specially because 5G and beyond systems are expected to experience an exponential increase in complexity, due to the massive deployment of SCs and the massive increase in the number of devices connected to the network [8, 11]. Furthermore, due to the static design of current mobile networks, operators are usually limited in network design options, and usually end-up designing their systems for the worst case scenario. If we take the previous example into consideration, the deployment of new network infrastructure in a building, often network engineers will ask questions such as how many people work in the building, or what are the peak hours of operation, or what time is lunch time. All of these questions contribute for network engineers to know what the maximum capacity of the new network should be, so that they can design and plan the system accordingly. However, this leads to extremely sub-optimal solutions, as whenever the building is not crowded or during off-peak hours, several network resources, as well as revenue, are wasted. If we just consider a simple case of day and night time, it is clear that this design is far from optimal, as if the deploy-

ment is happening in a commercial building, for example, the traffic experienced by the mobile network is usually much greater during the day than during the night [38].

Another example that clearly highlights the inefficiency of this one-off design approach is whenever certain one-off or sporadic events happen, such as in the case of natural disasters [39]. Due to the lack of adaptability and flexibility of current mobile networks, whenever the majority or even part of the network infrastructure is destroyed beyond repair, current mobile systems do not have the ability or the capacity to automatically recover from it, such as a self-reconfiguration or self-healing phase, for example [8,39]. As such, the remaining infrastructure cannot cope with the change experienced in the network, nor the affected structure can autonomously repair itself, leading to several users being without service, until a network operator comes and manually restore the network. Lastly, another example that clearly highlights the disadvantages of this fixed network design is when there is a large concentration of users in a single area, which can happen either in the case of natural disasters, as well as big events, such as music concerts, open markets or public holidays. In such cases, despite the network being configured to operate up to the worst case scenario, whenever one-off events happen, even in these situations, the current network infrastructure still cannot cope with the large increase in demand, leading to many users being without coverage. These last examples also point out to the problem of underprovisioning network resources, which can happen whenever operators deem that deploying a fixed infrastructure to cover a certain area might not be cost-effective to them (such as in temporary events, hot-spots, or rural areas) [39].

In addition to these issues, which are related to the fixed design of mobile networks, current solutions to network problems implemented by operators are also rudimentary [8]. Nowadays most solutions still rely on control loops, feedback controllers, or even a simple comparison against a threshold. All of these solutions require the constant collection of data and monitoring, which whenever an event or anomaly is detected, expert personnel will be deployed in order to try to find and solve the problem as soon as possible [8, 20]. As such, current methods are extremely ineffective, which often leads to a waste of network operators' money and subscribers [8, 11].

Although operators are slowly integrating SON capabilities in their networks, with some early SON functions being deployed in current LTE systems, current SON methods are still designed in a reactive manner [20]. Furthermore, current SON solutions generally assume that some information about the network is available, such as the location of coverage holes, handover ping-pong zones,

congestion spots or the location of users [20]. This occurs because in classic SON a delay is assumed, in which operators can observe the situation through alarms and data from their operations and monitoring centre, diagnose the situation and then trigger the compensating action utilizing any SON function. However, this approach cannot deliver the stringent requirements of future mobile networks and it would also not be suitable to construct dynamic and autonomous models of the network, nor make future predictions about the network operation [20]. Moreover, the assumption that operators have partial control or information about network events is also not completely realistic and do not contribute to the design of an autonomous and adaptable network, as it would still require human inputs [8, 9]. As such, in order to fully enable an autonomous, adaptable and agile network, other approaches that are capable of automatically analysing incoming real-time network data, determining what and where the problem is, and triggering specific actions should be designed, and for that to be possible, more intelligence needs to be incorporate in all layers of future networks [8].

2.1.2 5G and Beyond

The previous generations of mobile networks were designed with a clear goal in mind: to provide connectivity to end-users and make them able to communicate with each other. However, this inherit design of mobile networks has recently shown its limitations, as, nowadays, the majority of the traffic in mobile networks is based on video [8, 12–14, 16]. As such, the legacy design of mobile networks is not able to cope with certain requirements, such as in terms of bandwidth and capacity, as well as latency and reliability. Furthermore, with the advances in other technologies, it is expected that not only people will populate the next generation of mobile networks, but also machines [8]. As such, the legacy design of previous networks will be even more inefficient if applied to future mobile networks, as a wide range of requirements, not only based on data rate, will have to be met in order to enable all the future applications demanded by these new devices. Consequentially, it is clear that new paradigms are needed in future networks to address all the different requirements and use-cases. Based on that, a brief overview of new paradigms that are going to be present in 5G and beyond mobile networks and that can enable these new technologies is given below.

Ultradense Deployment

Network densification is considered a critical component of future mobile networks in order for them to cope with the expected exponential increase in traffic and

capacity. However, network densification will tremendously increase the burden on operators, as mobile networks will become increasingly more complex, with the configuration of thousands of parameters per BS, and the generation and collection of even more data than current networks [8,16]. Furthermore, it is also expected that the densification process will not be totally under the control of network operators, with end-users also contributing to the deployment of SCs in the network, such as in private offices or homes [8]. This, by its turn, will make life for operators tougher, as it will be harder to track all deployed BSs and configure them manually. Thus, solutions that can handle this increase in complexity and also deal with the unknown environments can be extremely advantageous in these cases.

Internet of Things and Machine-to-Machine Communications

The Internet of things (IoT) paradigm envisions that in the near future, everyday objects will be equipped with communications devices, which will enable them to communicate with one another as well as users and the core network, becoming an integral part of the Internet [40]. Moreover, by making everything connected, such as home appliances, city infrastructure, sensors, actuators, vehicles, etc., IoT and machine-to-machine (M2M) communications will enable a wide range of applications that are not possible today, such as home and industrial automation, telemedicine, smart cities and smart grid, and many others [26,40]. As such, with the exponential increase in the number of devices with Internet access, future networks will have to support billions of devices, with a variety of requirements and applications, such as bandwidth, latency and reliability, for example. As a result, 5G and beyond networks will have to be extremely efficient and diverse in order to attend all of these heterogeneous demands [41].

Shift from Reactive to Proactive

As seen in previous sections, current mobile networks are reactive instead of proactive, or in other words, operators wait for inputs from the network in order to analyse the data and determine if something went wrong in order to fix it [8]. However, this approach is not the most optimal one, as it can result in extensive periods of outage, when problems are not detected, poor QoS delivery and, subsequentially, loss of revenue. As such, a shift in paradigms is needed, in which future networks should become more proactive. One way of doing this is by considering more intelligent and robust algorithms, such as ML, in which previous historical data is analysed and predictions about the future state of the

network can be made [8].

Network Function Virtualization (NFV)

Another important enabling technology in future mobile networks is the concept of NFV, which aims at decoupling network functions from specific hardware components. This is done by performing these functions at high-end servers or in the cloud and then sending the result back to the hardware [8]. By decoupling functions from hardware, traditional issues encountered in previous generations of mobile networks can be overcome. Furthermore, centralized solutions, or solutions that would require additional resources, such as the huge amount of data generated by mobile networks, can more easily be implemented, as everything can be performed in a single location and then massively distributed to other devices [8].

Massive MIMO

The concept of massive MIMO aims to explore the current benefits of MIMO (in terms of capacity, reducing error rate - by diversity, increased spectral efficiency, etc.) in a much larger scale [42]. With the advances of the electronic industry, by being able to miniaturize components, according to the Moore's Law, future network systems could be coupled with arrays containing hundreds of antennas, simultaneously serving many user terminals [42]. As such, the concept of massive MIMO is expected to provide the required capacity enhancement needed, as well as improve the EE of future networks. However, several issues still remain in massive MIMO, specially in terms of interference management, downlink channel estimation, and physical layer challenges, such as waveform modulation and pilot contamination [43]. As such, adaptable solutions that can change communication parameters online can be a potential solution to some of these problems. One example can be the utilization of ML algorithms to overcome interference management, by dynamically adjusting the power of nearby BSs, or even supervised learning techniques in order to predict physical layer parameters and minimize network overhead [8].

Millimetre Waves (mm-Waves)

Lastly, another concept that has seen increased attention recently in future mobile networks, is the area of mm-waves. The utilization of mm-waves for signal propagation in wireless networks could be used to move away from the already saturated 700MHz to 2.6GHz radio spectrum bands that are currently widely

used in mobile communications. In addition to this benefit, moving to higher frequencies would also allow operators to allocate larger bandwidths, which directly translate to higher data rates and network capacity [44]. Given this significant jump in bandwidth and other new capabilities offered by mm-waves, future networks will be able to handle a much larger capacity than current networks, being an essential part of future systems. However, issues specially concerning path loss, LoS and signal attenuation are still being investigated in mm-waves communications. As such, similarly to massive MIMO, solutions that can adapt themselves in order to optimise network parameters online and solve these problems while the network is operational are advantageous in these situations, such as energy efficiency or beamforming [45, 46].

2.2 Machine Learning (ML) as an Enabler of SON

As it can be seen, all of these new paradigms that will be present in future mobile networks require some sort of optimization in real time or information from the network in order to operate at their optimal point. For example, in the case of the ultradense network deployment, ML can be utilized by network operators in order to estimate the traffic of certain locations and determine where and how many new BSs should be deployed in a certain area. Another example is in the case of IoT and M2M, in which ML can be used in order to learn communication patterns of machines. This can be done utilizing data from humans, and trying to predict what the behaviour of M2M will be or simply learning from machines themselves. Lastly, another example where ML can excel is in making future mobile networks more proactive, in which historic data from the network can be collected and analysed by ML models in order to predict what type and where failures are most likely to occur.

As it can be seen, the application of ML in future mobile networks has a limitless potential. Due to their inherit property of being able to analyse data, build models and make future predictions, ML is expected to play a major role in future mobile networks in order to enable, not only all of these functions, but also the SON paradigm. This, by its turn, will make future networks more reliable, efficient, cost-effective and manageable [8].

ML is the science of making computers take decisions without being explicitly programmed to do so [47]. This is done by programming algorithms that analyse a given set of data and try to make predictions about it. Or, in other words, the

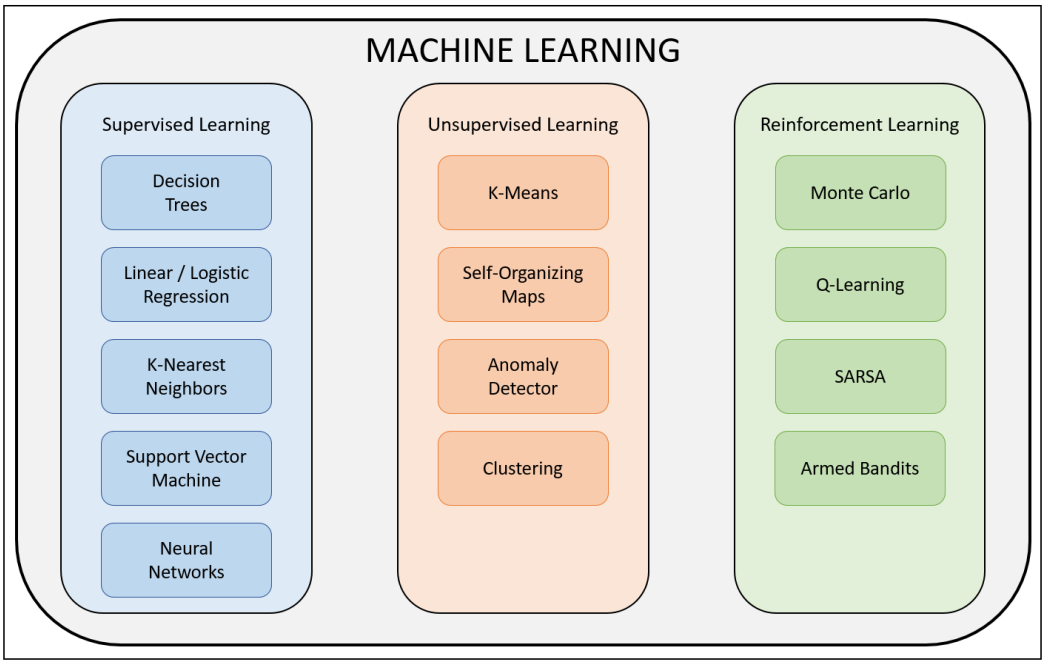


Figure 2.1: Block diagram showing the three main branches of ML and some of its algorithms [9].

goal of ML is to take an input data and learn a model based on a specific set of instructions (algorithm) that relates the given input with the desired output. Furthermore, depending on how learning is performed, ML algorithms can be classified differently, mainly into three categories:

- Supervised Learning;
- Unsupervised Learning;
- Reinforcement Learning (RL).

Below a brief description of each category is presented and Figure 2.1 shows how ML is divided into the three main branches and some of its algorithms.

2.2.1 Supervised Learning

Algorithms that require a data set that has information about both input and output data fall into the category of supervised learning. As the name suggests, this type of learning is similar to having a supervisor, or a teacher, supervising the learning process. The teacher knows the answers (output) for every input data, and, as the algorithm iteratively makes predictions during its training process, the teacher corrects it. More formally, supervised learning can be defined as algorithms that, based on the input and output relationship of the data, learn a

model that best represents the data and is able to make predictions for newly, unseen data examples [31, 32].

In addition, supervised learning algorithms are also split into two main categories, which varies according to the type of the output variable. If the output variable is a discrete variable, such as a class, for example: *Disease* and *No disease*, or *Cat* and *Dog*, the supervised learning problem is referred to as a classification problem. On the other hand, if the output variable is a real or a continuous value, such as the value of a house, or the birth rate of a certain region, then the supervised problem is considered as a regression problem [31]. Examples of algorithms from supervised learning range from very simple functions, such as linear regression, logistic regression, k-Nearest Neighbors, decision trees, support vector machines (SVM), to more complex ones, such as neural networks, and its variations, like convolutional neural networks and deep neural networks [33].

2.2.2 Unsupervised Learning

On the other hand, unsupervised learning algorithms are useful when the data set consists of data without information about its output [8, 48]. As such, these algorithms do not have the luxury of having a supervisor, and their objective mainly consists of discovering similarities in the data set and forming groups of similar examples, in what is known as clustering, or to determine the data distribution [31, 48]. Due to its nature of trying to estimate a model for data without labels, unsupervised learning algorithms consist mainly of grouping algorithms (clustering), such as K-Means, self organizing maps (SOM), anomaly detectors and mixture models [8].

2.2.3 Reinforcement Learning

RL is a unique type of learning, and quite different than supervised learning, which learns from previous examples provided by an external supervisor. Although important, supervised learning is able only to learn static models, and is not adequate for learning from interaction or from interactive problems, as in this type of problems, generating examples of desired behaviour are quite hard or even impractical to achieve [10]. As such, in interactive situations and in unknown territory (where one would expect intelligence to be the most beneficial) an agent must be able to learn from its own experience with the environment, thus, RL solutions are the most appropriate [10].

RL is a ML technique based on a goal-seeking approach [10]. In contrast to other ML techniques, such as supervised learning, in which the system learns

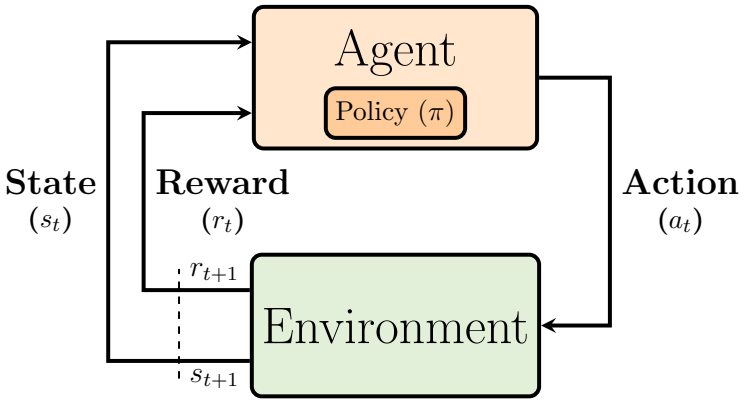


Figure 2.2: Block diagram of a RL system. The agent takes an action based on its current state. After taking an action, the agent receives a reward, which depends on the outcome of its action, and arrives in a new state.

by analysing examples provided by an external supervisor, whereas in RL, the learner must discover which actions to take by trying them [10, 49].

In RL, a system, called an *agent*, interacts with its surroundings, the *environment*. These interact continuously, with the agent selecting actions and the environment responding accordingly, by giving the agent a reward and presenting new situations, as shown in Fig 2.2. Basically, the agent and environment interact continuously at certain time-steps. At each time-step, t, the agent receives a representation of the environment's state and selects an action according to a policy (π). On the next time-step, $t+1$, as a consequence of its action, the agent receives a reward (r_{t+1}) and arrives in a new state. As such, the goal of an agent can be defined as to maximize its total cumulative reward. Based on this, a RL system can be divided into four main components [10]:

1. Policy (π): dictates the behaviour of the agent (how actions will be chosen) and it can be either deterministic or stochastic.

2. Reward (r_{t+1}): special numerical values given by the environment that the agent tries to maximize over time.

3. Value function: indicates the expected value of visiting a state, $V(s)$ – state-value function, or the value of taking an action in a specific state, $Q(s, a)$ – action-value function.

4. Environment: comprises everything outside the agent. In addition, usually the agent has some knowledge about the environment, however, sometimes, a model of the environment is not available.

Furthermore, due to the inherit nature of RL systems of exploring the environment and determining which actions to take, it is only natural that a trade-off

arises. This trade-off, known as the exploration-exploitation trade-off, is a fundamental dilemma in RL, in which the agent must not only exploit the best actions currently known, but also explore new actions, in order to determine if there are possible actions that lead to a better cumulative reward [10, 49]. In addition, RL algorithms can be divided into three main categories [10]:

1. Dynamic programming (DP): in which the agent has a perfect model of the environment, given by a Markov decision process (MDP), and the goal is to learn the optimal policy (in order to choose the best actions).

2. Monte Carlo methods (MCM): in this case, it is not assumed that there is complete knowledge about the environment. Thus, the agent must learn either online, by experiencing the environment, or through simulated experiences, in which the environment is represented by a very simple model.

3. Temporal-difference learning (TD learning): which can be defined as a combination of MCM and DP. Just like MCM, TD learning agents can learn directly from their experience with the environment, without the need of the complete environment dynamics. Furthermore, similar to DP, TD algorithms update their estimates (either a policy or a value function) based on other learned estimates (they are capable of bootstrapping).

These algorithms can be further divided into *On-Policy* or *Off-Policy*, depending on how learning is performed [10]:

- On-policy learning: the agent updates its value function and estimates the return (the total discounted future reward) assuming that the current policy continues to be followed.

- Off-policy learning: the agent updates its value function and estimates the return assuming a different policy than the one that is being followed.

In addition, one of the most commonly used policies in RL is the ϵ-greedy policy, which states that with a probability $p = (1 - \epsilon)$ the action that yields the maximum value known by the agent is chosen, whereas with probability $p = \epsilon$ an action is chosen at random. Furthermore, when a decaying ϵ rate is chosen, this policy exhibits a nice trade-off in terms of exploration and exploitation, in which in the beginning, because the ϵ is quite larger, the agent will favour exploring new actions, while later, due to the decaying ϵ rate, the agent favours the exploitation of the best actions [10].

Q-Learning

One of the most popular algorithms in RL is Q-Learning. First proposed by Watkins, in [50], Q-Learning is a TD Learning method that learns an action-value function, $Q(s, a)$, which represents the expected value of an agent being in a certain state and taking a specific action. Q-Learning is a method that, at each step at a state s_t, chooses an action a_t that maximizes its value function. This function, $Q(s_t, a_t)$, indicates how good is taking an action at a specific state according to a reward r. More formally, Q-Learning can be defined as [10, 49, 50]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \tag{2.1}$$

where $Q(s_t, a_t)$ is the current action-value function, α is the learning rate, r_{t+1} is the expected reward at the next time step, γ is the discount factor and $\max_a Q(s_{t+1}, a)$ is an estimate of the optimal future action-value function at the next time step, over all possible actions, a. In (2.1), the back-up, which is defined as what the algorithm stores in memory, is represented by the right side of the equation. Furthermore, the target is defined as $r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a)$, in which it represents the estimated value and the error is represented by $r_{t+1} + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t)$, which is the value that was previously stored in memory subtracted with the target value [10].

Because Q-Learning utilizes two different policies, one to generate its behaviour (ϵ-greedy for example) and another one that is evaluated and improved (fully greedy policy), it is considered to be an off-policy algorithm. One advantage of this separation is that the policy that is estimated can be deterministic, while the policy that controls the agent's behaviour can continue to sample all possible actions [10]. As it can be seen from (2.1), Q-Learning always evaluates the greedy policy (as it is the policy chosen to estimate the value of the action-function in the next time step), but as a behaviour policy, any policy can be chosen.

SARSA

Another commonly found algorithm in RL is SARSA, which stands for state-action-reward-state-action (SARSA). Similar to Q-Learning, SARSA also learns an action-value function, $Q(s, a)$, however, instead of choosing the next action based on the maximum expected value, SARSA chooses the next action according to the same policy used to choose the current action [10]. On the contrary to Q-Learning, SARSA utilizes the same policy to generate its behaviour and to evaluate and estimate the value of the action-function, thus SARSA is considered

to be an on-policy learning algorithm. Formally, SARSA can be defined as [10]:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]. \tag{2.2}$$

Similarly to Q-Learning, in (2.2), the back-up is represented by the right side of the equation, its target is defined as $r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1})$ and the error is represented by $r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$ [10]. As it can be seen, the only difference between SARSA and Q-Learning is the target value, in which in Q-Learning a the next action is chosen following a greedy policy, whereas in SARSA it is chosen according to the current policy being followed [10].

An example comparing the performance of Q-Learning and SARSA, as well as highlighting their differences, can be found in Appendix A.

Limitations in Conventional RL

In conventional RL, the task of learning value functions is often assumed to be the task of learning the values in a table with entries for different state-action pairs (also known as the Q-Table). However, despite this assumption working well in practice, the application of conventional RL algorithms are often limited to tasks with a small number of states and/or actions [10]. This occurs because RL algorithms rely on periodically updating lookup tables (the action-value function, for example) to determine which actions to take, which can be infeasible when the number of states or actions is too large [10].

Another issue commonly found in conventional RL algorithms is generalization, in which the algorithms are not capable of learning or estimating the value of the table between two very similar states. This happens because the basic idea behind many of the RL algorithms is to estimate the value function by using the Bellman equation as an iterative update [10]. Thus, the value-function is estimated separately for each sequence of states and actions, not being able to generalize for any given sequence. This is a severe problem, as in many tasks involving RL, most states encountered will never have been experienced before, specially in cases of continuous or complicated environments [10].

Thus, in order to overcome these issues, RL algorithms can be combined with generalization methods, such as function approximation, which aim to estimate the values of the tables by a function. By doing so, the tables can be represented by a particular function, in what is known as value function approximation (VFA).

Value Function Approximation (VFA)

VFA techniques aim to represent the value function of a RL problem not as a table, but as a parametrized function with a parameter vector $\vec{\theta}$. This means that the value of the function depends only on $\vec{\theta}$, varying from time steps as $\vec{\theta}$ varies. Typically, the number of parameters in $\vec{\theta}$ is much less than the number of states in a RL problem, as such, representing states become much less memory consuming [10]. Furthermore, because now the value function solely depends on $\vec{\theta}$, whenever a component of $\vec{\theta}$ changes, a change of estimates for many state-action pairs is triggered, consequently generalization is achieved [10]. As an example, one can think of the approximation function being a polynomial. Thus, whenever the parameters of the function (the coefficients, for example) change, it triggers a change in all estimated values. Furthermore, it is easy to see that if states with similar $\vec{\theta}$ inputs are input to the function, similar outputs are obtained. As such, it can be said that VFA brings generalization to RL, one key advantage that conventional RL does not provide.

In summary, the main idea behind VFA is to estimate the value function by experiencing and gathering examples from the environment. This can be achieved by interpreting the problem as a supervised learning problem, in which each backup of RL algorithms are seen as training examples, and the output is the observed target value for each point in the value function [10]. By doing this RL algorithms are able to use any existing function approximation methods for predicting the values of the value-function. Some examples of function approximators are: decision trees (DTs), linear or logistic regression, neural networks, as well as deep neural networks, which recently started the field of deep RL [34, 51–54]. This function is learned on-line by the agent interaction with the environment, and enables RL algorithms to generalize and estimate the values of new unseen states, while also addressing the problems related to memory required to fill in and update their tables [10]. However, despite these advantages, VFA introduces more complexity to the system, as now additional computation is necessary in order to estimate the value of the tables.

In order to perform learning in VFA, a certain function must have its parameters minimized, in order to update the learned weights, $\vec{\theta}$. As such, one technique that can be used to update these values is the gradient-descent (GD). GD methods are widely used in all function approximations and are particularly well suited for RL problems [10]. GD is an iterative optimization algorithm that finds the minimum of a function. In the case of RL, the function to be minimized can be expressed as the mean square error (MSE) between the target and the observed value function (or in other words, the error). For SARSA, for example,

the MSE can be defined as:

$$MSE(\vec{\theta_t}) = \sum_{s \in S} \left[r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]^2 . \tag{2.3}$$

In GD, it is usually assumed that the parameter vector $\vec{\theta_t}$ has a fixed number of real components, as in $\vec{\theta_t} = (\theta_t(1), \theta_t(2), ..., \theta_t(n))$, and that the value function is a differentiable function with respect to the parameter $\vec{\theta_t}$ [10]. As it can be seen from (2.3), it is also assumed that as the agent interacts with the environment, on each time-step t, a new example $Q(s_t, a_t)$ is observed. However, even if the agent knows the exact values of the value function, a difficult problem still remains, due to the limited resolution and resources of the function approximator. One such issue that arises is that it is generally impossible to find a parameter vector $\vec{\theta_t}$ that matches all the observed examples of the value function [10]. Moreover, even if such parameter vector existed, it would overfit to the observed distribution and it would not be able to generalize to examples that were not observed. As such, one strategy in order to overcome this issue is to minimize the error on the observed examples. This can be done by utilizing GD methods and updating the parameter vector $\vec{\theta_t}$ by a small amount in the direction that reduces the error the most on that example [10]. Considering SARSA, $\vec{\theta_t}$ can be updated as in [10]:

$$\vec{\theta}_{t+1} = \vec{\theta_t} - \frac{1}{2} \cdot \lambda_{\text{GD}} \cdot \nabla_{\vec{\theta_t}} \left[r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t, \vec{\theta_t}) \right]^2 . \tag{2.4}$$

By utilizing the chain-rule, (2.4) becomes

$$\vec{\theta}_{t+1} = \vec{\theta_t} - \frac{1}{2} \cdot \lambda_{\text{GD}} \cdot \left(-2 \nabla_{\vec{\theta_t}} Q(s_t, a_t, \vec{\theta_t}) \left[r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t, \vec{\theta_t}) \right] \right), \tag{2.5}$$

which can be re-written as

$$\vec{\theta}_{t+1} = \vec{\theta_t} + \lambda_{\text{GD}} \left[r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right] \cdot \nabla_{\vec{\theta_t}} Q(s_t, a_t), \tag{2.6}$$

where λ_{GD} is the GD learning rate, $[r_{t+1} + \gamma \cdot Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$ is the error in SARSA, and $\nabla_{\vec{\theta_t}} f(\vec{\theta_t})$ is the gradient for any function f, which is the vector of partial derivatives and is defined as [10]

$$\nabla_{\vec{\theta_t}} f(\vec{\theta_t}) = \left(\frac{\partial f(\vec{\theta_t})}{\partial \vec{\theta_t}(1)}, \frac{\partial f(\vec{\theta_t})}{\partial \vec{\theta_t}(2)}, ..., \frac{\partial f(\vec{\theta_t})}{\partial \vec{\theta_t}(n)} \right). \tag{2.7}$$

This partial derivative vector with respect to $\vec{\theta}$ points in the direction of the greatest rate of increase in any function, however, because the overall step in $\vec{\theta_t}$

is proportional to the negative gradient of the squared error, it will point to the direction that the error falls the most rapidly [10].

SARSA with VFA

VFA is a really powerful technique that enables RL algorithms to generalize and avoid the finite memory limitation of the system. One example of the application of VFA is in SARSA. Although SARSA and Q-Learning are very similar, because SARSA is an on-policy method, VFA has better convergence properties than Q-Learning, which is an off-policy method (in other words, off-policy algorithms tend to have convergence issues when VFA is applied). Based on the workflow of SARSA, an algorithm that showcases how VFA can be implemented is shown in Algorithm 1 [10].

Algorithm 1: SARSA with VFA [10]

```
for Every episode do
    Initialize current state
    for All actions do
        Get features present in current state and action
        Estimate value of Q-Table with VFA
    end
    Choose action according to policy
    for Each iteration do
        Take action
        Observe reward
        Move to next state
        for All actions do
            Get features present in next state and action
            Estimate value of Q-Table with VFA
        end
        Choose next action according to policy
        Update weights by GD
        Current state receives next state
        Current action receives next action
    end
end
```

2.3 State-of-the-Art in SON

In this section a literature review, covering state-of-the-art solutions that apply different ML techniques in the self-optimization function of SON, is presented. However, before that, a brief introduction to the functions of self-configuration and self-healing is given, and its main use-cases are highlighted. Due to the

broadness of such areas, and since the main focus of this thesis is in the application of RL algorithms in the context of self-optimization of SONs, a complete literature review covering these two areas is out of the scope of this thesis. However, for interested readers that would like to read a complete review on ML techniques applied in each use-case of every self-x function, as well as some suggestions on which ML algorithm to use depending on the use-case, please see [8].

2.3.1 Learning in Self-configuration

Self-configuration can be defined as the process of automatically configuring all parameters of network equipment, such as BSs, relay stations and femtocells. In addition, self-configuration can also be deployed after the network is already operable. This may happen whenever a new BS is added to the system or if the network is recovering from a fault and needs to reconfigure its parameters [11].

In order to perform self-configuration, several learning techniques are being applied to configure, not only basic operational parameters, but also to discover BSs neighbours and perform an initial configuration of radio parameters. However, due to the increasingly complexity of BSs, which are expected to have thousands of different parameters that can be configured (many with dependencies between each other) and the possibility of new BSs joining the network or existing ones failing and disappearing from their neighbours' lists, the process of self-configuration still provides quite a challenge for researchers.

Based on these steps, three major use cases of self-configuration can be defined and are divided into:

1. Configuration of operational parameters;
2. Determination of new BS neighbours and creation of neighbour cell list;
3. Configuration of the remaining radio related parameters and adjustment of network topology.

2.3.2 Learning in Self-healing

Current healing methods not only rely on manually interventions and inspection of cells, but also on reactive approaches, that is, the healing procedures are triggered only after a fault has occurred in the network, which degrades the network's overall performance and also results in a loss of revenue to operators [8].

The self-healing function in SON is expected not only to solve eventual failures that might occur, but also to perform fault detection, diagnosis and trigger

automatically the corresponding compensation mechanisms. In addition, it is expected that future mobile systems also move from a reactive to a proactive scenario, in which faults and anomalies can be predicted and the necessary measures taken before something actually happens. Due to this change in paradigm in current mobile networks, self-healing solutions are extremely challenging and rely heavily on previous gathered data in order to build models and try to predict whenever a fault might occur in the network [8].

From a learning perspective, several ML algorithms can be applied, depending on the type of data that operators have and its nature. In some scenarios, it is easy to label certain types of data, such as in fault classification, in others, however, such as in outage cases, in which outage measurements appear to be normal or only deviate a slight amount from normal, it might be more suitable not to label the data and work with unsupervised algorithms. Based on the collected references and also from [21], which defines the major use cases for self-healing, the following use-cases for self-healing could be defined as:

1. Fault Detection;
2. Fault Classification;
3. Outage Management.

2.3.3 Learning in Self-optimization

In SON, the concept of self-optimization can be defined as a function that constantly monitors the network parameters and its environment and updates its parameters accordingly in order to guarantee that the network performs as efficiently as possible [11]. Since the environment in which the network is inserted is not static, changes might occur and the BSs might need to adjust its parameters in order to accommodate the demands of the users. Changes can be in terms of traffic variations, due to an event happening in a certain part of a city for example, coverage, due to a network failure, capacity, because of a change in users mobility patterns, such as a road block or an accident, and many others [8].

Due to this fact, some of the initial parameters configured in the self-configuration phase might not be suitable any more and a change in can be required in order to optimize the network's performance. Since there are several different optimization parameters in the network, many ML algorithms can be applied. In addition, mobile operators also collect lots of data during network operation, which further enables the application of intelligent solutions in order to optimize the network. However, despite the huge amount of data collected, self-optimization is still a

challenging task, as many parameters have dependencies between them and a change in one of them can alter operation of the network as a whole [8].

Based on the use cases defined by [22] and the literature reviewed, SON use-cases in terms of self-optimization can be defined as:

1. Backhaul;
2. Caching;
3. Coverage and Capacity;
4. Antenna Parameters;
5. Interference Control;
6. Mobility Management;
7. Handover Parameters;
8. Load Balancing;
9. Resource Optimization;
10. Call admission control (CAC);
11. Energy efficiency (EE);
12. Coordination of SON functions.

Below, an overview of how ML algorithms are applied in each of these use-cases is presented.

Backhaul

One important aspect of future mobile network systems is the backhaul connection, or in other terms, the connection between the BSs and the rest of the network. Current mobile systems only evaluate the quality of the connection between the end-user and the BS. In the future, however, as systems will require to support a wider range of applications and heterogeneous requirements from users, this approach might not be suitable and a more end-to-end approach, considering the whole link between the user and the core network might be better [8]. With that in mind, some researchers developed solutions in order to solve the backhaul problem in future networks in terms of QoS and QoE provisioning [1–3, 55], congestion management [17, 56] and also topology management [57].

Solutions such as [17, 56] propose a backhaul solution involving flexible QoS schemes, congestion control mechanisms, load balancing and management features. In these solutions, the authors demonstrate a test-bed involving a network consisting of twenty nodes and with separated control and data plane. Another possible solution for backhaul optimization is proposed in [57], in which the authors utilize a fuzzy logic controller (FLC) to arrange the network topology in response to changes in traffic demand.

Other backhaul optimization solutions are the works proposed by Jaber et al. in [1–3, 55]. In these works the authors used Q-Learning to intelligently associate users with different requirements, in terms of capacity, latency and resilience, to SCs depending on the backhaul connection that they offered. If the backhaul and the user needs match, then the user would be allocated to that cell, otherwise a new cell is searched. Results showed that the proposed solutions are able to achieve better QoE for all users at the cost of decreasing overall throughput.

Caching

During the last couple of years, the fast proliferation of smart-phones and the rising popularity of multimedia and streaming services led to an exponential growth in multimedia traffic, which has very stringent requirements in terms of data rate and latency. In order to address these requirements and also reduce network load, specially during peak hours, future mobile networks must be coupled with caching functions. Some problems that arise, however, are the decision of what, where and how to cache, in order to maximize the hit-ratio of the cached content and provide gains to the network.

In [58], Wang et al. provide a good overview of why caching is necessary in future networks, what might be the gains of caching at different locations within the network and also presents some of the current challenges encountered. In terms of caching solutions, several approaches are being considered, such as in [59–64]. In [59], the authors explore various ways of integrating big data analytic with network resource optimization and caching deployment. The authors propose a big data-driven framework, which involves the collection, storage and analysis of the data and apply it to two different case studies. The paper concludes that big data can bring several benefits in mobile networks, despite of some issues and challenges that still need to be resolved.

Other caching solutions, like in [63], analyse the role of proactive caching in mobile networks. In this paper, the authors analyse and propose two solutions. First, the authors develop a solution to alleviate backhaul congestion, in which files are cached during off-peak periods based on popularity and correla-

tions among users and file patterns using collaborative filtering (CF). The second solution analyses a scenario that explores the social structure of the network and tries to cache content in the most relevant users, allowing a device-to-device (D2D) communication framework. These influential users, as they are called, would then have content cached into their devices and disseminate it to other nearby users. By using K-means algorithm, this second approach can cluster users and determine the set of influential users and which users can connect to them.

Another approach from the same authors as in [63] is shown in [61]. In this work the authors apply a new mechanism based on transfer learning (TL) in order to overcome the problems of data sparsity and cold-start problems that can be encountered in CF. In this new solution, the authors assume that they have gathered data and built a model for a source domain, composed of a D2D based network. After that, the proposed TL solution smartly borrows social behaviours from the source domain to better learn the target domain and builds a model that can smartly cache contents into the BSs.

Other solutions for caching optimization include the work in [60,64], where the caching problem is modelled as a game theory problem. In [64], the authors model the system as a many-to-many matching game and propose an algorithm that is capable of storing a set of videos at BSs in order to reduce delay and backhaul load. On the other hand, Blasco et al., in [60], tackle the optimization problem of storing the most popular contents in order to relieve backhaul resources. Another work that researched the impact of caching in mobile networks is [62]. In this solution the authors propose the optimization of caching in SC networks and divide it into two sub-problems. First, a clustering algorithm (spectral clustering) was utilized in order to group users with similar content preferences. After that, RL is applied so that the BSs can learn which contents to cache and optimize their caching decisions.

Coverage and Capacity

Another challenging issue in future network systems is the optimization of coverage and capacity, in which the network tries to optimize itself in order to achieve the best trade-off between coverage and capacity. Based on this, several authors are proposing intelligent solutions to tackle this problem [8].

In [65], for example, the authors apply SOM (unsupervised learning) to optimize the number of cells inside a cluster and also antenna parameters in order to achieve a better coverage. Two different scenarios are proposed, the first one aims to change only cluster sizes, while the second one changes both cluster sizes and

antenna parameters. On top of that, two SOMs are considered to perform cluster optimization. It is shown that the first scenario provides a gain of around 5%, while the second one achieves a gain in the order of 13%. Other approaches, such as in [66–68], utilize feedback controllers in order to optimize the coverage and capacity of the network. In [66], a coverage adaptation mechanism for femtocell deployments that utilizes information about mobility events of passing-by and indoor users to optimize femtocell coverage is developed. Fagen et al., in [67], propose a method to simultaneously maximize coverage while minimizing the interference for a desired level of coverage overlap. On the other hand, Engels et al., in [68], develop an algorithm that tunes transmit power and antenna down-tilt angle in order to optimize the trade-off between coverage and capacity via a traffic-light based controller.

Furthermore, the work in [69] considers a novel multi-objective optimization (MOO) model and proposes a meta-heuristic approach in order to perform coverage optimization. The solution simulated a LTE network scenario and aimed to maximize the performance of users in a given cell in terms of fairness and throughput. Other solutions, such as in [70, 71], attempt to optimize the coverage of femtocells by using GAs. In both solutions, the authors tried to perform a multi-objective evaluation and the algorithm would try to satisfy three rules simultaneously: minimize coverage holes, perform load balancing and minimize pilot channel transmit power. In the end, the solution returns the best individual of all populations and changes the pilot power of femtocells accordingly.

Antenna Parameters: Another set of parameters that also have an impact on coverage and capacity of the network are antenna parameters, mainly: antenna down-tilt and azimuth angles, and transmit power. In particular, the optimization of antenna parameters often requires tuning after the initial operator's configuration and are very delicate, requiring not only expertize, but also a lot of precision to perform. Hence, it can be quite costly for the operators to perform this optimization and that is why several papers are trying to automatically optimize the antenna's parameters.

In [72], the authors propose four different methods in order to optimize traffic offload of macrocells to microcells. The first two solutions utilize only microcell measurements, while the third method is based on minimization of drive test (MDT) measurements and the last method is a hybrid of all three previous solutions. All methods, however, aim to maximize capacity offload from macrocells, or in other terms, maximize microcells' coverage. By changing the antenna down-tilts and transmission powers according to the measurements collected via a

feedback loop mechanism this offload is achieved. In [73], the authors develop an optimization algorithm to find the best settings for antenna down-tilt angle and common pilot channel power of BSs. The solution begins by performing an evaluation of the network and analysing the obtained results. After that, an iterative process formed by a control loop begins. In this process, parameters are changed according to certain rules and how far the parameters are from optimal until an accepted level is reached. Other works, such as in [74–76] aim to optimize the down-tilt angle of the antennas by applying fuzzy Q-Learning (FQL) in a LTE network scenario in order to achieve better coverage. While in [77], Eckhardt et al., propose an algorithm for antenna down-tilt angle optimization in order to optimize the spectral efficiency of users. The approach considered a LTE network scenario and is based on heuristics to find the best antenna parameters.

Interference Control: interference has always been a problem affecting the performance of communications systems and in future networks this will not be different. Hence, several intelligent approaches are being considered in order to cope and control this limiting factor.

In [78], for example, the authors propose a distributed self-organizing femtocell management architecture in order to mitigate the interference between femtocells and macrocells. The solution consists of three feedback controllers, in which the first loop controls the maximum transmit power of femtocell users, the second determines each femtocell user's target signal to interference plus noise ratio (SINR) and the third attempts to protect the users uplink communication. Another approach is [79], which proposes a feedback controller to perform intercell interference cancellation (ICIC). The proposed solution consists of two phases: in the initial phase, each cell attempts to assign resources by itself and, in the second phase, cells optimize themselves by resolving sub-optimal assignment of the resources. It is shown that the algorithm is capable of achieving good results and also assign resources reliably.

Mehta et al., in [80], develop two solutions in order to address the problem of co-layer interference (interference between neighbours) in a heterogeneous macrocell and femtocell network scenario. The two schemes attempt to mitigate co-layer interference while also improving the minimum data rate achieved by femtocell users and ensuring fairness to them. The first scheme proposes a modification to the technique of adaptive frequency reuse (AFR) by adding power control, while the second scheme applies a self-organized resource allocation solution based on a feedback controller in order to allocate resources and manage the interference. In [81], the authors also build a self-configuration and optimization

scheme for a network of femtocells overlaid on top of a macrocell network. The algorithm automatically configures the femtocells transmit power and promotes self-optimization via a feedback controller to automatically control when to turn on or off femtocells in order to reduce interference between macro and femtocells.

Other approach to interference mitigation is the work in [82]. In this work, the authors model the coexistence of a macrocell and femtocell network and develop a distributed algorithm for femtocells to mitigate their interference towards the macrocell network. The authors divide the problem into two sub-problems of carrier and power allocation. The carrier allocation problem is solved via Q-Learning, while the second sub-problem, of power allocation, is solved using a gradient method. Another solution that utilizes the concept of RL, is the work in [83], in which a solution to the problem of ICIC in the downlink of mobile orthogonal frequency-division multiple access (OFDMA) systems is proposed. The problem is posed as a cooperative multi-agent control problem and its solution consists of a fuzzy inference system (FIS), which later is optimized using Q-Learning. The solution is based on the concept of adaptive soft frequency reuse and the ICIC concept is presented as a control process that maps system states into control actions, which can be modelled as a RL system. Lastly, another solution comes from Aliu et al., in [84], in which the authors adopt a novel fraction frequency reuse (FFR) based on GA for ICIC in OFDMA systems. The main difference of this solution is that it not only attempts to use a new technique, but also considers a non-uniform distribution of users and characterizes it by determining its centre of gravity.

Mobility Management

Another important aspect of future mobile network systems is the ability to predict user's movement in order to better manage resources and reduce the cost of network functions, such as HO. Mobility management can be defined as the process in which the network is able to identify in which cell the user currently is [85]. Current location techniques involve databases that store the locations of the users and every time the user changes position these databases need to be updated [86], however, this method is not very efficient. If the network could predict a user's next cell or even the path it will traverse, several gains in the network performance could be observed, hence, different solutions are being developed to this challenging problem.

Some papers, such as in [85–92] use back propagation neural betworks (BPNNs) in order to predict the next cell a user can be. The basic idea behind all these papers is to use the concept of NN to learn a mobility-based model for every user

in the network and then make predictions of which cell the user is most likely to be next. In [89], for example, the authors develop a method consisting of two cascaded ML models. The first model performs clustering via K-means while the second does classification. In classification, the authors compare the performance of three different methods, mainly, NN, DTs and naive Bayes. Results show that the proposed model achieves better accuracy than performing only classification alone and also that the classifier that performed the best was the DT classifier. Despite using NNs as primary intelligent strategies, some papers also use different learning techniques. Akoush et al., in [88], combine the concept of NN with Bayesian learning in order to perform classification tasks and predict a user's next cell and show that Bayesian networks outperform standard NN by 8% to 30%.

Another supervised learning technique that can be found in the mobility use-case is the support vector machine (SVM). In [93] Chen et al. build a model that uses only channel state information (CSI) and HO history to determine a user's mobility pattern. Their algorithm defines an user trajectory based on the previous and next cell it traversed and, given the input data (previous cell and CSI sequence), the next cell can be predicted. In addition, the solution considers multiple classifiers, one for each possible previous cell, and trains several non-linear SVM classifiers with Gaussian kernels. On the other hand, authors from [94] consider the problem of estimating not only the location of mobile nodes in an indoor wireless network, but also channel noise. The solution uses a Hierarchical SVM model, composed of four different levels and is able to maintain good accuracy for speeds up to 10m/s. Other approaches to mobility prediction are the works in [95, 96], in which the authors propose a movement prediction and a resource reservation algorithm, which uses MC and hidden Markov models (HMMs), respectively. In [95], the authors considered a discrete-time MC in order to represent cell transitions and determine a user's path. This approach does not require any training and optimization is done online. On the other hand, the solution of [96], models the network as a state-transition graph and converts the problem into a stochastic problem. HMM is then applied, so that it learns the mobility parameters and, later, makes its predictions. Lastly, another solution that relies on the use of MC is the work in [97], in which the authors propose a movement prediction and a resource reservation algorithm. The movement prediction algorithm is done via distributed MC while bandwidth management is done in a statistical way.

In another set of solutions, this time from Sas et al., in [98, 99], the problem of users that have high mobility and experience frequent HOs is addressed. The algorithm shown in [98] consists of three major components, a trajectory classifier,

trajectory identifier and a traffic steerer. The objective of the algorithm is to classify and match current trajectory of users with previous trajectories stored in a database. After that, the steerer is activated so it can decide if it is better to keep the user in the current cell or to perform a HO. The solution in [99] builds upon that and adds a mobility classifier module before the steerer makes a decision. By implementing this classifier, the algorithm becomes more generic and can determine in which categories users fall into, e.g.: slow, medium or high mobility, before deciding if they need to be steered or not.

On the other hand, Yu et al., proposed a novel approach based on activity patterns for location prediction in [100]. Instead of predicting directly a user's next location, the solution attempts to, first, infer what the user's next activity is going to be, to, later, predict the location. The approach consists of three phases. The first phase tries to infer the current activity that the user is doing, the second attempts to infer the next activity and the third predicts the location. The proposed algorithm uses a supervised model to build an activity transition probability graph, which also takes into account the variation of time, so at different times of the day, the activities predicted by the model might be different, as it should be.

The work proposed in [101] attempts to use semi-supervised or unsupervised techniques to reduce the effort of gathering labelled data to perform location prediction. To perform this, the authors build a discrete model and assign a Gaussian distribution to model the signal strengths of received signals by users for every location. After that, two different approaches are taken. In the first approach, the authors label only part of the data, making it a semi-supervised model, while in the second approach a data set with no label is considered. After that, the authors learn a model and use it to compute the estimate of location for each test sample. The authors conclude that there is significant opportunity to explore semi-supervised and unsupervised learning techniques since even without any labelled data (fully unsupervised model), a reasonably accuracy could be obtained.

Recently, a work by Farooq et al., in [102], proposes the use of a semi-Markov model together with participatory sensing in order to predict mobility pattern of users in the network. Lastly, another recent work, is the work proposed by Mohamed et al., in [103], in which the authors build upon the previous model presented in [95]. By using an enhanced MC to predict next cell locations for users of the network, the authors demonstrate that by predicting a user's next location HO signalling costs can be reduced.

Handover Parameters Optimization

The process of changing the channel (frequency, time slot, spreading code or a combination of them) associated with a connection while a call is in progress is known as hand over (HO). HO are of extreme importance in mobile networks due to the nature of mobility of its users. Without this procedure, mobility could not be supported as connections would not survive the process of changing cells. HO can be divided into two categories, there can be horizontal HO (HHO), in which a user switches between BSs of the same network or vertical HO (VHO), in which a user switches between BSs of different networks. The optimization of HO parameters is crucial in many aspects of the network, as it can affect not only the mobility aspect, but can also affect coverage, capacity, load balancing, interference management, and energy consumption to name a few [8].

Furthermore, the tuning of HO parameters also has an influence in several other metrics used by operators which are important to determine if the network is performing well, such as ping-pong rate, call dropping probability, call blocking probability, and early or late handovers [104]. Due to its importance, a substantial amount of research is being done in this area and several ML approaches are being considered. In [105], for example, the authors discuss the impact that changing the A3-offset, and time to trigger (TTT) parameters or the application of certain techniques, such as mobility estimation or cell range extension offsets (CREOs) can have in the HO procedure. The authors also propose a solution for the mobility robustness optimization (MRO) case and demonstrate the performance gains of CREO in a heterogeneous network scenario. Other authors, such as Soldani et al., in [106], propose a generic framework for self-optimization and evaluate the impact of pruning NCL in terms of HO.

One possible solution to the optimization of HO parameters, can be in terms of NN, as seen in [107–109]. In [107], for example, the authors develop a new HO algorithm based on probabilistic NNs and compare it with the current hysteresis method. Results show that the NN reduces the number of HOs performed, reducing the cost of signalling of the whole network. On the other hand, authors from [108, 109], propose algorithms to optimize the HO procedure and better determine when an user needs a HO. Another technique utilized in order to optimize the HO procedure is SOM. In [110], Sinclair et al. develop a method to optimize two HO parameters, hysteresis and TTT, and achieve a balance between unnecessary HOs and call drop rate. The proposed algorithm has a different view from the main solutions, as it is more interested in which cell to tune the parameters, rather than how to tune them. Also, their model is based on a modified version of SOM, XSOM, which allows for a kernel method to replace the distance measure-

ments of SOM, allowing a non-linear mapping of inputs to a higher dimensional space. Results show that the XSOM solution is able to reduce the number of dropped calls and unnecessary HOs by up to 70%.

On the other hand, Stoyanova et al., in [111], propose two different methods to solve VHO optimization. The first method is based on a FLC and involves measuring certain metrics, like: signal strength, bit error rate, latency and data rate in order to vote pro or against the HO for each mobile terminal. While the second approach involves the use of SOM, in which a few parameters (same as previous method) are periodically measured and, each of them, independently, can cause a HO initiation. Results show that the fuzzy solution performs really well and allows a simultaneous evaluation of different HO criteria. Unfortunately, the same cannot be said for the SOM solution. The authors conclude that SOM might not be appropriate for HO decision-making.

Another class of algorithm that is widely used in HO optimization is the class of feedback controllers, as can be seen from [41, 104, 112–122]. All of these solutions aim to change HO parameters, such as hysteresis, TTT, A3-offset, HO margins, cell offsets or stability periods based on the measure of performance metrics and how far they are from optimal. FLCs are also widely used in the context of HO optimization, as it can be seen in the works of [18, 123–130]. All of these algorithms consists of gathering certain network related metrics, fuzzifying them and making decisions in order to optimize HO margins, thresholds, hysteresis, TTT, or other attributes, so that the network can make better HO decisions.

Other solutions proposed for the optimization of HO parameters are in the context of RL. Mwanje et al., in [131], develop a distributed Q-Learning solution for the MRO use case. The contribution of the paper lies on the fact that their solution is able to adjust HO settings (hysteresis and TTT) in response to mobility changes in the network. The solution in [132] also relies on Q-Learning. This time, however, the authors consider both MRO and mobility load balancing (MLB) use cases. In the MRO solution, the primary goal is to determine optimal HO settings, while in MLB the objective is to redistribute load between cells. Another solution to the HO optimization problem is the work of Quintero et al. in [133]. In this paper, a hybrid GA solution is considered in order to solve the problem of assigning BSs to radio network controllers (RNC) in a third Generation (3G) network scenario. Another approach that uses GAs, is the work in [134], in which the authors propose a solution that enables every cell of a LTE network to adjust its HO parameters (HO margin, A3-offset and TTT), in order to minimize call drop and unnecessary HOs. In [135], Bouali, et al., propose an algorithm based

on a FLC combined with a fuzzy multiple attribute decision making methodology in order to choose which network should a user connect to, depending on the the users' application and its requirements.

Another solution to HO management is proposed in [136], in which the authors utilize two NNs in order to determine which cell should an user handover to based on the user's perceived QoE in terms of successful downloads and average download time. Dhahri et al., in [137], propose a cell selection method for a femtocell network. In this work three different approaches for cell selection are considered, first a distributed solution is proposed, secondly, a statistical solution is presented and the third solution relies on game theory. By determining which cell users should connect, the algorithm is able to maximize the capacity and minimize the number of HOs for every user of the network. Another work, [138], also by Dhahri et al., proposes two different approaches for a cell selection mechanism in dense femtocell networks. The algorithms rely on Q-Learning and FQL and try to optimize, based on previous data, the best performing cell in the future for each user in the system. Results show that the enhanced FQL outperforms conventional Q-Learning and that the algorithms are capable of reducing the number of HOs while also maximizing capacity.

Load Balancing

In order to cope with the unequal distribution of traffic demand and to build a cost-efficient and flexible network, future networks are expected to balance its load intelligently [8]. One solution, proposed in [139], aims to enable a heterogeneous LTE network to learn and adjust dynamically the CREO of SCs according to traffic conditions and to balance the load between macro and femtocells. The algorithm utilizes a regression method in order to learn its parameters and then uses its model to adjust the CRE offsets.

Another approach involves the use of feedback controllers, such as in [140,141]. In [140], the authors propose a solution to modify HO thresholds in order to decrease the served area of overloaded cells and increase the area of underloaded cells and hence, achieve load balance. Similarly, in [141], the authors also develop a solution based on the control of HO parameters. This time, the goal is to find the best HO offset values between an overloaded cell and a possible target cell. Rodriguez et al., on the other hand, propose the use of a fuzzy controller to achieve load balancing in LTE networks, in [142]. The authors also implement a FLC in order to auto-tune the HO margins to balance traffic and reduce the number of calls blocked. Muñoz et al., also propose the optimization of HO parameters to achieve load balancing in [143] by combining the concepts of FLC

and Q-Learning in a second generation (2G) network scenario. Another similar work, is shown in [144]. This time, however, the authors investigate the potential of different load balancing techniques, by tuning either transmission powers or HO margins, to solve persistent congestion problems in LTE femtocells. The paper proposes solutions based only on FLC and also FLC combined with Q-Learning. Results show that the strategy that considered Q-Learning performed better and also performance gains were larger when Q-Learning was applied to optimize transmission power instead of HO margins. Another approach that uses the concept of Q-Learning is the work by Mwanje and Thiel in [145], in which CREOs are adjusted between a source cell and all its neighbours by a fixed step. Then, a Q-Learning is applied to determine the best step value for every situation. The authors show that the new method performs better than a fixed-step solution. Another work that explores Q-Learning is [146], in which a scheme is built so that every user learns to which cell to send a service request in order to reduce the number of outages and also achieve load balancing.

Other solutions, such as in [147–149], attempt to solve the load balancing problem in a heuristic way. In [147], the authors develop an algorithm to balance unequal traffic load while also improving the system performance and minimizing the number of HOs. The algorithm relies in a greedy distributed solution and considers a LTE network scenario. In [148], the authors propose a load balancing method by creating clusters dynamically via two different methods, centralized and decentralized heuristics. Lastly, the work of Al-Rawi, in [149], studies the impact of dynamically changing the range of low power nodes, by applying CRE. The solution aims to enable femtocells to take users from macrocells by adding a CRE offset to the received signal power of the users. Results show that dynamic CRE benefits the majority of users in the network, but does this by trading-off gains from picocell to macrocell users.

In [150], the authors propose a dynamic sector tilting control scheme by using GAs to achieve load balancing. The solution aims to optimize sector antenna tilting to change both cell size and shape to maximize the system capacity. Another solution is the work in [151], in which an approach is considered to balance load among neighbouring cells of the network. The algorithm consists of five different parts in which it analyses and determines which BS needs to have its traffic handled and determines to which neighbour to switch it to. The proposed method analyses historical data collected by the algorithm, if available, and predicts which neighbour should have its antenna down-tilt angle changed and by how much. Otherwise, if no data is available, a heuristic search for the best neighbour is performed. Lastly, a recent work proposed by Bassoy et al., in [152],

present an unsupervised clustering algorithm in a control/data separation plane. Results show that the proposed solution is able to offload traffic from highly loaded cells to neighbour cells and that the algorithm can work in a high dense deployment scenario, making it suitable for future mobile networks.

Resource Optimization

Another important aspect of future networks is the optimization and provisioning of resources. One example is the work in [59], in which the authors explore various ways of integrating big data in the mobile network. In this paper, the authors propose a big data-driven framework and analyse use cases in terms of resource management, caching and QoE. All solutions are based on the collection and analysis of data in order to better determine how the network can change its parameters. The authors conclude by stating that big data can bring several benefits to future networks, however there are still significantly challenges that need to be solved.

Some solutions, like the ones proposed in [153–159] rely on the use of NNs in order to optimize network resources. In [153], Sandhir and Mitchell develop a scheme that predicts a cell demand after every 10 measurements taken by the system. At each prediction interval, the predicted resource usage in each cell is compared with the number of free channels available and channels are reallocated between cells, with the ones having more channels giving to the ones having less channels. In [154] user mobility is predicted by using two NN models in order to reserve network resources in advance. In [159], Adeel et al. build a cognitive engine that analyses the throughput of mobile users and suggests the best radio parameters. The solution relies on the application of a random NN and three different learning strategies are investigated: GD, adaptive inertia weight particle swarm optimization (AIW-PSO) and differential evolution (DE). The authors show that AIW-PSO performs better and also converges faster. Zang et al., in [155], propose a method based on spatial-temporal information of traffic flow using K-means clustering, NN and wavelet decomposition to predict traffic volumes on a per cell basis and allocate network resources accordingly. Another solution that applies NN, is the work in [158]. This time, however, the authors use a regression based NN and aim to predict the path loss of a radio link, in order to optimize the BSs transmission power. Another solution is shown by Railean et al. in [156], in which an approach for traffic forecasting is proposed by combining stationary wavelet transforms, NN, and GA. The paper adopts several different approaches based on similarity between days and also training of the NN and results show that when GAs were applied the performance decreased. Similarly,

the work in [157], also develops a traffic forecasting solution and has as primary goal to determine voice traffic demand in the network.

In [160], Binzer et al. builds a self-configuration mechanism that determines the number of BSs needed in the network and also a self-optimization technique in order to optimize BSs location and antenna parameters. The algorithm relies in a SOM solution in order to move BSs accordingly and minimize the total number of under and oversupplied points in the network. Kumar et al., in [161], propose a game-theoretic approach in order to optimize the usage of resource blocks in a LTE network scenario. The solution uses a harmonized Q-Learning concept and attempts to share resource blocks between BSs. Savazzi and Favalli, in [162], build two novel approaches for downlink spatial filtering based on K-means clustering algorithm. The first method groups users in clusters using K-means algorithm and then computes beam widths by considering the power level of edge users. The second method also uses K-means clustering, but after that, it compares for each user the best BSs available. Based on this, users might be reassigned to different BSs and overall system capacity can be increased. Another approach is the work in [163]. In this work, a Q-Learning based algorithm is proposed in order to adjust femtocells power, so that the capacity of the network was maximized maximize while interference levels were maintained within certain limits. In addition to Q-Learning, the paper also develops a TL solution between macrocells and femtocells, in which macrocells would communicate their future intended scheduling policies to femtocells. By doing this, the femtocells can reuse the expert knowledge already learned for a certain task and apply it to a future task.

In [164], the authors propose a cluster and feedback loop algorithm to perform bandwidth allocation. This algorithm explores user and network data in order to increase overall throughput. Kiran et al., in [165] develop a Fuzzy controller combined with big data in order to find a solution for bandwidth allocation in RAN for LTE-A and 5G networks. On the other hand, Liakopoulus et al., in [166], build an approach to improve network management based on distributed monitoring techniques. Their solution monitors specific parameters in each network BS and also considers that BSs interact with each other. Due to this interaction, BSs can take self-optimizing actions based on feedback controllers and improve network performance. In [167], the authors propose a framework for fractional power control (FPC) for uplink transmission of mobile users in a LTE network. The solution utilizes a FLC combined with Q-Learning in order to reduce blocking rate and file transfer times. Another solution that also utilizes Q-Learning is [168], in which the authors develop a scheme to maximize resource utilization

while constrained by call dropping and call blocking rates. Their solution can achieve performances comparable to other classical methods, but has the advantage of not requiring explicit knowledge of state transition probabilities, like in Markov solutions.

Call Admission Control (CAC): CAC is a function of network systems that tries to manage how many calls there can be at a certain time in the system. Basically, if a new call comes to the network, either by someone making a new call or by transferring a call from another cell (via HO), this function determines if that call can be admitted or not in the system based on how many resources are available at that current time. Based on this, it can be said that CAC regulates access to the network and tries to find a balance between number of calls and the overall QoS provided, while also trying to minimize the number of dropped and blocked calls.

Several works have been published covering the optimization of CAC, such as: [169–177]. In [170], for example, the authors propose a CAC function that relies not only on information about the system resources, such as available bandwidth, but also on predictions made regarding system utilization and call dropping probability. By constantly monitoring these parameters and using a feedback controller, the authors are able to predict if a call should be accepted or rejected by the system for two different type of traffic classes, voice traffic and multimedia traffic.

Other authors, such as [172–177] rely on the use of FLCs in order to perform their CAC algorithm. Most of these solutions rely on estimating a set of parameters, such as effective bandwidth and mobility information in [173, 176], cpu load in [177], or queue load in [175], to determine whether to accept or reject a call. On the other hand, the work in [171], propose a different approach to solve the CAC problem. In this work, the authors utilize a generic predictor scheme (in this case a Markov-based scheme) integrated with a threshold based statistical bandwidth multiplexing scheme in order to perform CAC for both active and passive requests. Based on the predictions given in terms of user mobility and time of arrival and permanence time, the algorithm then makes its decision.

Another approach to CAC is developed in [169], in which a RL solution is built in order to tackle the problem in a code division multiple access (CDMA) network. The solution involves four steps in order to work. First, data is collected and calls are either accepted or rejected based on any CAC scheme available. After that, the RL network is trained and then the trained network is applied to the simulated scenario. Lastly, the fourth step consists of updating the network via

a penalty/reward mechanism. Results show that the proposed method achieves better performance in terms of grade of service (GoS).

Energy Efficiency (EE): another problem that arises with the network densification process is the increase in energy consumption of the network. To overcome this issue, which would cut operators costs and also enable a greener network, several intelligent solutions are being developed [8].

One possible solution is proposed by Alsedairy et al., in [18], in which a network densification framework is introduced. However, instead of deploying regular SCs, the authors exploit the notion of cloud SCs and fuzzy logic. These cloud cells are smart cells that underlay the coverage area of macro cells and, instead of being always on, they communicate with the macrocells to become available on demand. As such, by optimizing the availability of SCs, the network can reduce its overall energy consumption.

Zhao and Chen in [81] also propose a mechanism to promote EE in the network. Their solution relies on a feedback controller in order to determine when to turn on or off a femtocell. Similarly, in [178], the authors build a scheme to dynamically activate or deactivate modular resources at a BS, depending on the network conditions, such as traffic or demand. The approach involves a RL algorithm, based on Q-Learning, that continuously adapt itself to the changes in network traffic and makes decisions of when to turn on an additional BS module, turn off an already activated module or to maintain the same condition. The proposed solution can achieve a very high energy saving, with gains of about 80% without increasing user blocking probability.

In [179], Peng and Wang apply an adaptive mechanism to increase the standardized energy saving mechanism (ESM) quality. The framework relies on adjusting sleep intervals of cells based on network load and traffic. The algorithm relies on the concepts of MC and can save network power and also guarantee spectral efficiency. The solution divides the energy saving process in three scenarios, heavy, medium and light loads, and, for each scenario, the adaptive solution is investigated. The authors conclude that the proposed adaptive solution is better than standard solutions, specially in light loads scenarios, while in higher loads, both schemes achieve similar performances. Another solution is presented in [180], in which the issues of improving traffic load and network planning are tackled. Their solution first builds supervised prediction models in order to predict traffic values and then applies the information gathered from external planned events in order to improve prediction quality. Based on the traffic demand prediction, the framework is then able to turn on or off certain cells in the network, achieving

EE.

Recent work by Jaber et al., in [55], tried to intelligently associate users with different BSs depending on their backhaul connections. In the proposed scenario, each BS had multiple backhaul connections and an energy optimization, in terms of which backhaul links to turn on and off, was performed. Another recent solution is the work proposed in [181] by Miozzo et al., in which Q-Learning was used in order to determine which BSs to turn on or off and to improve the energy usage of the network. Lastly, the work in [182] utilizes big data, together with supervised learning (polynomial regression), in order to optimize the energy of ultra dense mobile networks. The authors show that the proposed solution can achieve the highest cell throughput while maintaining EE, when compared to conventional approaches.

Coordination of SON functions

Another important issue that arises with the advent of SON is how to coordinate and guarantee that two or more distinct functions will not interfere with each other and try to optimize or adjust the same parameters at the same time [183]. One simple example of this can be a hypothetical scenario where the network tries to minimize its interference level, but at the same time it tries to maximize its coverage. To avoid this type of situation, it is essential that SON functions are coordinated to ensure conflict-free operation and stability of the network.

Lateef et al., in [183], develop a framework based on DT and policies in order to avoid conflicts related to the mobility functions of MLB and MRO. Another important contribution of the paper is that it classifies the possible SON conflicts into five main categories, mainly: key performance indicators (KPIs) conflicts, parameter conflicts, network topology mutation, logical dependency conflicts and measurement conflicts. Another approach that tries to resolve the SON conflict management is proposed in [184]. The authors consider a distributed coordination scenario between SON functions and analyse the case in a LTE network scenario. Each SON function can be viewed as a feedback loop and are modelled as stochastic processes. The authors were able to conclude that coordination is essential and that it can provide gains to the system.

Other solutions involving feedback controllers, can be seen in [185, 186]. In [185], the authors start by presenting a hybrid classification system of SON conflicts. The authors state that, since many SON conflicts can fall into more than one category, this hybrid approach is better and propose a fuzzy classification to accomplish that. The authors also evaluate some use-cases of SON conflicts and present distributed solutions based on feedback controllers, in which mea-

surements are gathered, evaluated and the parameters changed accordingly. Similarly, in [186], Karla also classifies SON parameters, but his classification is only based on the parameters impact on the mobile radio system, resulting in only two classes of parameters. Karla also presents a proof of concept scenario, in which a simplified LTE-A scenario is simulated and coordination is evaluated. First, the system performs a set of offline computations in order to find good configuration parameters and then the system uses a feedback controller to update itself in an online manner.

Table 2.1 shows a summary of the reviewed papers for the self-optimization use cases and how they are distributed in terms of ML techniques.

Table 2.1: Self-optimization use cases in terms of ML techniques

		Backhaul	Caching	Coverage and Capacity	Mobility	Handover	Load Balancing	Resource Optim.	Coord.
Sup. Learning	**Bayes**	-	-	-	[88, 89]	-	-	-	-
	NN	-	-	-	[85–92]	[107–109, 136]	-	[153–159]	-
	SVM	-	-	-	[93, 94]	-	-	-	-
	DT	-	-	-	[89]	-	-	-	[183]
	CF	-	[63]		-	-	-	-	-
	Misc.	-	-	-	[100]	-	[139]	[180, 182]	-
Unsup. Learning	**K-Means**	-	[63]	-	[89]	-	-	[155, 162]	-
	SOM	-	-	[65]	-	[110, 111]	-	[160]	-
	Game Theory	-	[60, 64]	-	-	[137]	-	[161]	-
	Misc.	-	-	-	[101]	-	[152]	-	-
Controller	**Feedback**	-	-	[66–68, 72, 73, 78–81]	-	[41, 104, 106, 112–122]	[140, 141]	[81, 164, 166, 170]	[184–186]
	FLC	[57]	-	-	-	[18, 111, 123–130, 135]	[135, 142, 144]	[18, 165, 172–177]	-
Reinf. Learning	***Q*-Learn.**	[1–3, 55]	[62]	[82, 83]	-	[131, 132, 138]	[145, 146]	[55, 163, 168, 178, 181]	-
	FQL	-	-	[74–76]	-	[138]	[143, 144]	[167]	-
	Misc.	-	-	-	-	-	-	[169]	-
Markov	**MC**	-	-	-	[95,97,103]	[137]	-	[171, 179]	-
	HMM	-	-	-	[96]	-	-	-	-
	Misc.	-	-	-	[102]	-	-	-	-
Heuristic	**Heur.**	-	-	[69, 77]	-	-	[147–149, 151]	-	-
	GAs	-	-	[70, 71, 84]	-	[133, 134]	[150]	[156]	-
TL	**-**	-	[61]	-	-	-	-	[163]	-
Misc.	**-**	[17, 56]	[59]	-	[98, 99]	-	-	[59]	-

Chapter 3

Backhaul Optimization using RL

3.1 Introduction

As previously mentioned, 5G networks will have to accommodate a huge amount of users with a wide range of requirements. This can be an issue in future networks, specially in terms of user-cell association, as current association approaches are centred in two major assumptions:

- The radio interface is the bottleneck of mobile networks;
- There is little variation in user requirements.

As such, these two assumptions will render static association approaches in 5G networks completely ineffective. Thus, due to the rising challenges of 5G networks and heterogeneous demands of multiple users, a new paradigm must be designed for the association of users and BSs. Furthermore, new technologies, such as network densification and new air interfaces, are shifting the bottleneck of future mobile networks to the backhaul [1, 187–189]. However, due to the sub-optimality of current cell association methods, the backhaul of the associated cell might not be sufficient to satisfy specific user needs, and as such, more intelligent approaches that consider the end-to-end connection (RAN and backhaul) and users requirements is needed [1].

As such, a user-specific cell association algorithm is envisioned in order to tackle the problem of allocating users with distinct requirements to the best fitting SCs with different backhaul parameters. Based on that, a solution which considers an end-to-end approach and attempts to associate specific users with specific cells is proposed. The framework aims to tune both SCs cell range extension offsets (CREOs) and user requirements weights, with the objective of determining the best combination that satisfy each user. If that is not possible, however, the algorithm attempts to find the best combination that minimize

users' dissatisfaction. Users can have different needs in terms of network parameters, such as throughput, latency, resilience, energy efficiency, or security, while each SC has certain attributes associated with these parameters as well. The main idea and innovation behind the proposed RL based algorithm is, to perform two different optimizations, one at the network level, in which the algorithm optimizes the CREO of SCs via Q-Learning, followed by another optimization at the user level, in which the algorithm determines the best weights for each user, also via Q-Learning. Combining both Q-Learning solutions and optimizing of both network and user parameters, the proposed solution is able to provide user-specific allocation and achieve better results in terms of user satisfaction and QoS.

3.1.1 Related Work

Since the main bottleneck of future mobile networks is expected to shift from the RAN to the backhaul, its optimization and the BS association problem have gained increased attention recently. In [188], the authors optimize the backhaul and BS assignment problem using a novel heuristic algorithm. However, only user throughput was considered as a QoS metric. Moreover, due to its heuristic nature, the proposed solution might not be computationally feasible, as it must determine for every network configuration, the parameters of all users and BSs. Olmos et al., in [189], build upon [188], and consider a more generic model based on Markov chains to solve the problem of cell selection with backhaul constraints. Despite being more general, the authors do not consider user QoS requirements and both [188, 189] do not consider a heterogeneous network scenario, in which BSs have different transmit powers and backhaul characteristics.

In [190], a method to balance network load of BSs backhaul based on their geometric location is proposed. In [191], the authors aim to optimize the user-cell association in a decoupled uplink and downlink heterogeneous network scenario. However, both [190,191], do not consider user QoS requirements when performing cell association. In [190], for example, the authors attempt to perform backhaul load balancing, while in [191] the reference signal received power (RSRP), cell load and backhaul capacity were regarded as limiting factors in cell association. In [192], the authors optimize the network backhaul throughput by improving the cell association process. However, as the authors mention, conventional search algorithms would not work for this problem, as the cell association problem is non-deterministic polynomial-time hard (NP-hard), becoming infeasible for a large number of users an BSs. Thus, they propose a heuristic centralized algorithm to associate users. However, [192] also does not consider users with different QoS

requirements, nor cells with different backhaul links. Pantisano et al., in [193], address the the cell association issue by considering that SCs can cache content in order to overcome backhaul capacity limitations and improve users QoS. However, for the proposed solution to work, user location must be known (or estimated) and only user throughput was considered as a QoS requirement.

Han et al., in [194], aim to optimize user association and resource allocation in a heterogeneous network considering radio resource consumption, energy and backhaul constraints. However, because the problem is NP-hard, the authors propose decomposition methods to reduce the problem to a smaller version, and to build an online solution. Also, the authors considered optimizing only the resource allocation and utility of the network via cell association and did not attempt to improve user QoS. In [195], network frame design, resource allocation and user association optimization in a heterogeneous massive MIMO network scenario is proposed. Although this solution can adapt to different network scenarios, it does not investigate user QoS requirements and only optimizes total network sum rate.

Ma et al., [196], investigate the user association and resource allocation in a massive MIMO heterogeneous network scenario and attempt to maximize network utility. The authors develop an analytical solution and despite considering a heterogeneous network scenario, users QoS requirements is not considered. On the other hand, Lee, et al., in [197] address the user cell association problem considering backhaul load balancing and minimizing user call blocking probability. The problem is formulated as 0-1 integer problem, but due to its complication it is relaxed to become a convex optimization problem. Lastly, works by Jaber et al., [1–3], propose an algorithm based on Q-Learning to solve the cell association problem by considering backhaul limitations. The proposed distributed solution aims to tune the CREO of SCs so that users can connect to the SCs with the backhaul that would better match their needs.

However, due to the analytical or heuristic aspect of these solutions, [188–197] may not be adaptable enough in order to enable future mobile network paradigms, such as SONs, as they often require unrealistic assumptions. Most of these works require the knowledge of how many users and BSs or SCs are in the network, or user positions and requirements, for example. Also, as it could be seen, almost none of the reviewed works consider user QoS requirements or the utilization of different applications and backhaul links. Furthermore, as these works often depend on searches or analytical expressions, periodical optimizations are often required and no network or user data is utilized, not fully exploiting the potential of SON. Moreover, as future networks are expected to be much more intelligent

and adaptable, by using historical or online data, solutions that do not require lots of assumptions and that can learn intrinsic patterns in data as the network changes are preferred. Thus, more general solutions that can analyse data and take online decisions, such as machine learning, should be designed [8].

Thus, a distributed RL based solution is proposed. Due to the inherit nature of RL solutions, a machine learning technique based on a goal-seeking approach [10], a model-free solution to the problem of user-cell association is proposed. In this case, differently from [188–197], no assumptions or prior knowledge are necessary, as all the data needed for the algorithm to learn is generated online by the network and its users. As such, the proposed method is more robust and generic, as it can adapt to different network conditions, while the previously reviewed literature require previous knowledge about their environment and are limited by the specific application designed to fit the solutions.

Other works, such as [1–3], also utilize RL to perform cell association, however the main drawback of these solutions is that only network parameters are optimized and user parameters and requirements are assumed to be random, achieving a sub-optimal solution. In [1–3], for example, it is assumed that users weights are binary random and do not depend on users QoS requirement. As such, the weights associated for each user and its requirements would not always conform with its demands leading to a limited optimization. Furthermore, this assumption can lead to network resource wastage, as users that did not have a stringent demand, could end up having high weights, while more demanding users could be assigned low weight. As such, the works in [1–3] do not optimize user parameters, but only network parameters (SCs CREOs), leading to a network centric (or BS-specific) solution and are not capable of solving the problem for each user individually. In addition, in [1–3], the proposed solutions are denoted as user-centric because the metrics evaluated are considered from a user perspective, but they do not actually perform any user optimization.

Thus, unlike [1–3], in which user weights are assumed to be random, the envisioned solution proposes an optimization of both network and user parameters, so that user-specific cell association can be achieved, leading to better network resource consumption and user satisfaction. Moreover, as future mobile networks are expected to be more user-oriented and deal with several applications with different requirements, it is only natural that solutions which try to optimize individual user and network parameters are developed. In addition, not only will different types of users need to be addressed, but also the same user could have different requirements at different times of the day, as it utilizes different applications. Hence, a solution that can adapt itself to different user needs and that

can treat users differently based on their current requirements is needed, and, for that to be possible, an intelligent user-specific approach is necessary.

3.1.2 Objectives and Contributions

As seen from the literature review, backhaul-aware cell association has been a focal topic of research in the recent years. However, solutions in this area still remain network-centric and agnostic to the diversity of user requirements. On the other hand, BS-centric association has been studied in the past in the works of Jaber et al. [1–3], which endows the cell association process with the ability to distinguish and prioritize users requirements. However, these efforts are still limited by the network parameter tuning and do not account for the users' ability to improve its choice. The proposed solution is the first to address this issue of both tuning network and user parameters and also to develop a two-step association scheme that outperforms prior state-of-the-art solutions with minimum added complexity.

The solution is based on RL, more specifically Q-learning, and it is shown to be robust and flexible to enable an autonomous cell association, enabling the stringent requirements needed for future mobile networks in a heterogeneous and diverse environment. By optimizing both network and user parameters, the proposed solution is able to allocate to each user what it needs without wasting network resources and making other users suffer. This, in its turn, enables more users to be allocated to the network while also satisfying their needs, improving individual and, by consequence, overall QoS. The contributions can be summarized as follows:

- To provide an end-to-end paradigm in terms of downlink user-cell association, considering the radio access network, backhaul conditions and users QoS requirements;
- To optimize the user-cell association process by delivering to each user only what was requested, minimizing network resource wastage;
- To perform both network and user parameters optimization, considering both network constraints and user requirements to achieve user-specific cell association, in an adaptable and intelligent manner via RL (Q-Learning).

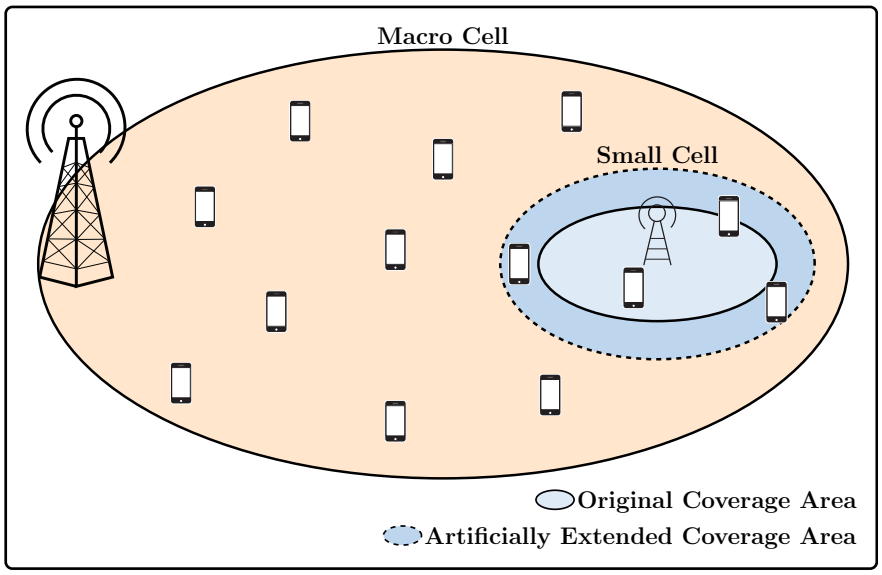


Figure 3.1: Example of how applying CREO can change the cell association of users in the network.

3.2 Background

3.2.1 Cell Range Extension Offset (CREO)

Current downlink cell association methods rely solely on radio interface parameters, such as the RSRP or SINR, to determine which cell a user should be associated to. In the future, however, as the next generation of mobile networks are expected to be much more diverse and heterogeneous by nature, the bottleneck of mobile networks will shift from the RAN to the backhaul and current association methods will probably not be adequate [1, 2, 188, 189, 198]. Since the transmit power of a macro BS and a SC are very different, much in favour of the macro BS, the problem of load imbalance in the network is created. If only the RSRP or SINR is considered, most users would prefer to connect to the macro BS, as it has a higher transmit power, overloading it and leaving the SCs underloaded [199, 200].

To solve this issue, a technique known as CREO was developed, in which SCs artificially extend their coverage area by adding an artificial offset to the user perceived RSRP in the association process [201]. Figure 3.1 shows an example, in which, a SC is overlaid on top of a macro cell. Initially, the SC covers only the light blue area, however, when CREO is applied, the SC's coverage area is artificially extended to the darker dotted area. Hence, users within this greater coverage area will now prefer to connect to the SC instead of the macro cell.

Although adding a CREO can provide several benefits, such as enhanced

uplink data rates, increased capacity (by means of load balancing), and improving network robustness, by making SCs less sensitive to their deployment location, only artificially increasing the perceived transmit power of SCs is not enough to improve the performance of the network [201]. If a fixed CREO was applied to all SCs, for example, the problem of load balance would be solved, but the problem of backhaul congestion would be created. Also, since tuning SCs CREOs only considers the problem from a RAN perspective, this would not completely attend different users requirements and would not be able to provide enough QoS, nor meet the requirements for future networks [1–3].

Hence, it is clear that optimizing only the radio interface is not enough, and that a joint optimization between the radio interface and the backhaul is needed. In addition, SCs should also adjust their CREOs more efficiently and intelligently, so that users with different requirements can connect to the SCs that best fit their needs. Also, if only the parameters from the network side are tuned, only a group of users can be satisfied, as those with the highest priority to a specific requirement [1,2]. Thus, in order to provide a user-specific cell association, which attends to a wide range of user requirements, it is clear that an optimization at the user side must also be done, so that users can be intelligently associated to the best fitting SC.

3.3 System Model

A downlink heterogeneous network scenario is considered. In this scenario, a single macro cell is positioned in the center and is divided in m sectors, denoted as $\mathbb{M} = \{M_1, ..., M_m\}$. On top of each sector, n SCs, $\mathbb{S} = \{S_{M_1,1}, S_{M_2,1}, ..., S_{M_m,n}\}$, are randomly distributed, and the set with all cells in the system is defined as $\mathbb{C} = \mathbb{M} \cup \mathbb{S}$. Also, each SC has a single non-ideal last-mile connection, while the backhaul connection between the macro BS and the core network is assumed to be ideal. In addition, each SC is assumed to have η adjustable CREOs, one for each backhaul parameter. These offsets are defined as $\mathbb{O} = \{O_{1,1}, ..., O_{c,\eta}\}$, with $c \in [1, |C|]$, and each CREO can assume values from $\mathbb{V} = \{V_1, V_2, ..., V_{max}\}$.

On top of this network, $\mathbb{U}$ users are distributed and each sector is considered to have k users, with higher concentration near the SCs, $\mathbb{U} = \{u_{1,1}, u_{m,2}, ..., u_{m,k}\}$, and $p = m \cdot k$ is the total number of users. Furthermore, each user has μ required parameters ($\mu = \eta$), which can be seen as QoS parameters that a user is concerned about. In addition, users QoS requirements are represented by $\mathbb{E} = \{E_{u,1}, ..., E_{u,\mu}\}$, and, for each requirement, each user has an associated weight to it (defined by the application, for example), denoted as $\mathbb{W} = \{W_{u,1}, ..., W_{u,\mu}\}$.

These weights are adjustable, and can assume values in $\mathbb{G} = \{G_1, G_2, ..., G_{max}\}$ and the network monitors the QoS of users, represented by $\mathbb{E}' = \left\{E'_{u,1}, ..., E'_{u,\mu}\right\}$.

It is assumed that each user can connect to a single cell at a time and cells have limited radio and backhaul resources. Whenever a user is allocated to a BS, it consumes one resource block (RB) and both macro and SCs have a limited amount of RBs. This assumption is made for comparison purposes and, as such, the resulting gains are not defined by the number of RBs each user has, but by finding a more suitable cell association[1]. Regarding interference, it is assumed that macro and SCs share the same frequency band, and a frequency reuse factor of one is considered. In addition, enhanced intercell interference coordination (eICIC), a technique in which two BSs communicate with each other to mitigate the effects of interference, is also assumed.

Based on this eICIC framework, almost blank subframes (ABS) are implemented. ABS is a technique introduced in 4G, to enable the coexistence of different BSs. In this technique, system that share spectrum band negotiate parameters in order to guarantee a fair coexistence [1–3, 7]. In this case, a time-sharing mechanism is implemented, in which whenever the macro BS is transmitting the small cells are assumed to be silent and vice-versa. Lastly, as per 3GPP current standards, the CREO of serving and neighbouring cells are broadcast to users in the vicinity using common control channels.

3.3.1 Cell Association

In order to associate users to cells, the received signal power from each cell is computed. The RSRP, $R_{u,c}$, (in dB) of user u and cell c, can be expressed as

$$R_{u,c} = P_c - 10 \cdot \log_{10}(N_{sc}) - H_{u,c} - L, \ \forall \ c \in \mathbb{C}\,, \tag{3.1}$$

where P_c is the transmit power of cell c, N_{sc} is the total number of sub-carriers[2] in cell c, $H_{u,c}$ is the path loss between user u and cell c, and L is the penetration loss.

A log-distance path loss is assumed and is defined as [202]

$$H_{u,c} = \Psi + 10\gamma \cdot \log_{10}\left(d_{u,c}\right) + X_\sigma\,, \tag{3.2}$$

where Ψ is a propagation constant, γ is the propagation exponent, $d_{u,c}$ is the distance between the user u and cell c, and X_σ is defined as the log normal

[1]This is assumed for the sake of simplicity, but, in practice, RB allocation could be done dynamically.

[2]Sub-channels in a specific time-slot are considered (definition of a RB).

shadowing component.

Based on the received power from each cell, users are then going to decide which cell to associate with. This is done by a ranking system, which takes into account only the perceived RSRP, if the user is trying to connect to the macro cell, or the RSRP combined with the SC's CREO and user weights, in case it is a SC. The cell ranking can be expressed as

$$K_{u,c} = \begin{cases} R_{u,c}\,, & \text{if } c \in \mathbb{M}. \\ R_{u,c} + \frac{1}{\eta} \sum_{i=1}^{\eta} W_{u,i} \cdot O_{c,i}\,, & \text{if } c \in \mathbb{S}. \end{cases} \tag{3.3}$$

After each user ranks every cell, the cell association process begins. If a BS has enough space to accommodate a user request, and the user's SINR is above a certain threshold, then the user connects to the desired cell. The perceived SINR of user u, and cell c, can be calculated as

$$\text{SINR}_{u,c} = \begin{cases} \frac{R_{u,c}}{N + \zeta_{ABS} \sum\limits_{i=1, i\neq c}^{n} \omega_i R_{u,i}}, & \text{if } c \in \mathbb{M}. \\ \frac{R_{u,c}}{N + \zeta_{ABS} \sum\limits_{i=1, i\neq c}^{n} \omega_i R_{u,i} + (1-\zeta_{ABS}) R_{u,M}}, & \text{if } c \in \mathbb{S}. \end{cases} \tag{3.4}$$

where N is the noise power, ζ_{ABS} corresponds to the fraction of ABS time that the SCs transmit (in percentage — between 0 and 1), ω_i is the load of SC i, $\sum\limits_{i=1, i\neq c}^{n} R_{u,i}$ is the RSRP from other SCs belonging to the sector that the user is connected to and $(1 - \zeta_{ABS})R_{u,M}$ is the interference from the macro cell sector that the user belongs to, scaled down by the percentage of time that the macro cell is not transmitting due to ABS.

If the BS does not have enough RBs or if the SINR is not high enough, then the user tries to connect to the next best cell. This process is repeated for the next four BSs until a connection can be established, or if that is not possible, the user is then assumed to be out of coverage in that time slot [1]. If however, a BS has more than enough RBs to serve its users, the remaining RBs are assumed to be unused during that time slot. If a user is connected, then the maximum user throughput is estimated as

$$T_{u,c} = B \cdot \log_2 \left(1 + \text{SINR}_{u,c}\right), \tag{3.5}$$

where B is the bandwidth occupied by one RB. In addition, the amount of back-

haul throughput required for all users connected to a cell is computed as

$$\lambda_c = \rho_c \cdot \sum_{u=1}^{U_c} T_{u,c}\,, \tag{3.6}$$

where $\rho_c > 1$ represents the backhaul overhead [1], and U_c denotes the total number of users connected to the cell.

Since SCs have limited backhaul capacities, whenever their required backhaul throughput exceeds its total capacity, the effective throughput of all users connected to that cell is reduced. The effective throughput of users is expressed as

$$T'_{u,c} = \begin{cases} T_{u,c}\,, & \text{if } \lambda_c \leq C_c\,, \\ T_{u,c} - \frac{C_c - \lambda_c}{U_c}\,, & \text{if } \lambda_c > C_c\,, \end{cases} \tag{3.7}$$

where C_c is the maximum backhaul capacity of cell c.

The throughput of each cell, c, can be calculated as

$$T_c = \sum_{u=1}^{U_c} T'_{u,c}\,, \tag{3.8}$$

and the total throughput of the system can be determined by

$$T = \sum_{c=1}^{|\mathbb{C}|} \sum_{u=1}^{U_c} T'_{u,c}. \tag{3.9}$$

3.4 Proposed Scheme

The objective of the proposed system is to maximize the total effective cumulative throughput of all users, given a set of constraints. This can be done by tuning both CREOs of SCs and user weights in a centralized manner. However, centralized solutions can be impractical, as it would require an extra layer of communication between the SCs, users, and the centralized unit in order to disseminate changes in the network, increasing signalling overhead. In addition, synchronization would become an issue, due to the centralized entity not being updated all the time. As such, using an outdated values fetched from the centralized unit could impact the performance of the system. Based on that, a distributed approach is preferred.

The proposed solution aims to divide the global optimization problem of maximizing the total system throughput into smaller sub-problems. These sub-problems can be defined as maximizing the throughput of each individual cell of the system, given certain backhaul constraints. More formally, the optimization

objective can be formulated as

$$\underset{\mathbb{O},\,\mathbb{W}}{\text{maximize}} \quad T_c\left(\mathbb{O},\mathbb{W}\right) \tag{3.10a}$$

$$\text{subject to} \quad \mathrm{RB}_c \leq \mathrm{RB}_{max}, \tag{3.10b}$$

$$C_c \geq \lambda_c, \tag{3.10c}$$

$$\sum_{u=1}^{U_c} \frac{E'_{u,\mu} - E_{u,\mu}}{E_{u,\mu}} \leq \theta_\mu,\ C_c \geq \lambda_c,\ \forall\ \mu, \tag{3.10d}$$

$$\frac{E'_{u,\mu} - E_{u,\mu}}{E_{u,\mu}} \geq 0,\ \forall u \in \mathbb{U},\ \forall\ \mu. \tag{3.10e}$$

where θ_μ represents a threshold that determines how much over satisfaction, on average, is allowed for each parameter μ. Note that constraints (3.10d) and (3.10e) have their signal changed when latency is considered (latency value is minimized, while the other parameters are maximized).

As it can be seen, maximizing the throughput of each individual cell of the network, in (3.10), is subject to four different constraints. The first constraint (3.10b) states that each cell c is limited by a maximum number of RBs, or in other words is limited to the number of users it can serve. The second one, (3.10c), states that each cell has a maximum backhaul capacity and if the total capacity required by the users associated to cell c exceeds it, then the throughput of all users associated to that cell is reduced, as defined in (3.7). In addition to this constraint, each SC keeps track of how many RBs it has, so the CREO optimization takes into account both radio and backhaul parameters. The third constraint, (3.10d), states that the average satisfaction level of users connected to a cell must be below a certain threshold, given that there are backhaul resources available in the SC. This constraint attempts to limit the amount of over-satisfaction users can have and aims to distribute backhaul resources better. By respecting this constraint, the system perceives allocating few users with too much of a certain resource as a bad manoeuvre, as more users would be left starving. Hence, the system will try to find a better user-cell association in order to reduce this over-satisfaction and distribute the backhaul resources better. It is also important to note that this constraint does not deal with resource allocation, it only attempts to satisfy users by changing the cell each user is associated with, specially in idle mode. For example, if a user requires 5ms latency, but is associated with a SC that provides 1ms of latency, this association is not very efficient, as this user is over-satisfied and is wasting resources that could serve other users that require lower latency. Thus, changing association of this user to a SC with higher latency would be more efficient, as the user could still be satisfied and the pre-

cious latency resource is freed for a more demanding user. Based on that, the value of θ_μ can then be chosen as a system parameter, which determines how much over-satisfaction, on average, is allowed at the expense of less satisfaction of other users. However, by considering only these two constraints, as in [1–3], only the aggregate performance of users connected to a certain cell is optimized, making the system not able to track individual user performance.

Based on these issues of dealing only with the aggregate performance of users, a fourth constraint, (3.10e), is proposed. This constraint states that each user should be allocated more than its target QoS (each user should be satisfied). It should be noted that constraints (3.10d) and (3.10e) have opposing optimization objectives. Consequently, satisfying both constraints results in a solution where each user measures a QoS value that is as close as possible to its target $E'_{u,\mu} \rightarrow E_{u,\mu}$. In other words, each user should be allocated only enough of each resource, so that it is satisfied. By doing this, the system guarantees that each user is satisfied, while enabling more backhaul resources to be shared, avoiding the limitation of being constrained by the aggregate performance of users as in [1–3], and achieving a user-specific solution.

In order to accomplish the objective defined in (3.10), a formulation based on RL is proposed, consisting of two different optimization processes. First, an optimization from the network perspective of SCs CREOs is performed. In this optimization, the SCs learn the best set of CREOs, $O_{c,\eta}$, that satisfies the majority of their users (this optimization addresses constraints (3.10c) and (3.10d)), similar to the optimization performed in [1–3]. After that, each user will optimize its own weights, $W_{u,\mu}$, also via a RL formulation, and as highlighted in the introduction, this is the main contribution of the paper, achieving a user-specific cell association.

3.4.1 SCs Learning

In order to solve the optimization problem in (3.10), an intelligent and distributed solution based on Q-Learning is proposed. The SCs belonging to $\mathbb{S}$ have a set of η adjustable CREOs, that can be learned in order to maximize the perceived throughput of each SC. Hence, each SC is considered to be an agent and the network is the environment.

The actions, a_c, that each SC can take are defined by the changes in their CREO values, $O_{c,\eta}$, described by $\mathbb{V}$. In addition, each SC is considered to have η attributes, and one adjustable CREO for each attribute. Each CREO is learned and adjusted independently from one another (each SC considers independent state-action pairs for each parameter, η). The policy that the agents follow in

order to take actions is a completely greedy one, in which the best action is chosen at every iteration. In terms of states, each SC can be in one out of three possible states:

- State 1, if constraint (3.10c) is not satisfied (the backhaul is currently overloaded).
- State 2, if constraint (3.10c) is satisfied and (3.10d) is not (the backhaul has resources available, but users have not been associated in an optimal way, as there are users over-satisfied).
- State 3, if both constraints (3.10c) and (3.10d) are satisfied — the SC can accommodate more users (its backhaul is not overloaded) and the user association is good enough.

More formally, the states, υ_c, that each SC can be are

$$\upsilon_c = \begin{cases} 1, & \text{if } \lambda_c > C_c\,, \\ 2, & \text{if } \sum_{u=1}^{U_c} \frac{E'_{u,\mu} - E_{u,\mu}}{E_{u,\mu}} > \theta_\mu \mid C_c \geq \lambda_c\,, \\ 3, & \text{otherwise.} \end{cases} \tag{3.11}$$

For each state-action pair a reward, r_{υ_c,a_c}, is associated, and it can be seen as a value corresponding to the consequence of taking certain action and ending up in a specific state [10]. The reward in this case is defined as

$$r_{\upsilon_c,a_c} = \begin{cases} A_1, & \text{if } \upsilon_c = 1\,, \\ \sum_{u=1}^{U_c} \frac{E'_{u,\mu} - E_{u,\mu}}{E_{u,\mu}}, & \text{if } \upsilon_c = 2\,, \\ A_2 \cdot \frac{C_c - \lambda_c}{C_c}, & \text{if } \upsilon_c = 3. \end{cases} \tag{3.12}$$

In State 1, as a cell should always try to avoid having its backhaul overloaded, a low reward (e.g. $A_1 = -1000$) is defined. In State 2, however, despite the backhaul of the SC not being overloaded, the association performed is not the best, as some users are over-satisfied (indicating that other users might be starving). Since this state is also not ideal, but not as bad as State 1, a low reward based on the percentage difference between what the majority of users achieved and requested is assigned. Lastly, in State 3, which is the best possible state a cell can be, the reward is defined as the percentage difference between a cell's maximum and current backhaul capacity, multiplied by a constant (e.g. $A_2 = 100$), and

represents how many more users are able to fit in cell c. A constant is added so that whenever a cell moves from one state to state 3, the algorithm will yield a high reward value.

Based on that, for each state-action pair and its reward, each agent learns and updates its η Q-Tables. Since these tables depend only on the state-action pairs, each Q-Table, $Q_{c,\eta}$ is an $[a_c \times \upsilon_c]$ matrix, and, for each iteration of the algorithm, they are updated following the formulation in (2.1). Lastly, since the algorithm operates in an iterative manner, it is only natural that a stopping criteria is devised to guarantee the convergence of the proposed solution. In this case, two stopping criteria are formulated. The first guarantees that the optimization is not perform indefinitely, as such, the SCs adjust their CREOs for a maximum number of iterations (M_{sc}), while the second states that if the reward does not improve from one iteration to the other more than a threshold, (r_{th}), it is also accepted that the algorithm has converged.

3.4.2 User Weights Learning

After the SCs learn their CREOs, which represent the best offset that will please the majority of the users connected to each cell, the user weights learning begins. Each user learns the weights, $W_{u,\mu}$, given to each parameter, μ, also using a Q-Learning formulation. In this learning problem, each user is considered an agent of the system and the network represents their environment. Each user can take certain actions, a_u, represented by changes in their weights $W_{u,\mu}$, described by $\mathbb{G}$ and the same greedy policy from the SCs learning is assumed. For each parameter, μ, each user can be in one of two states:

- State 1, if the user is not satisfied with respect to parameter μ (constraint (3.10e) is violated).
- State 2, if the user is satisfied with respect to parameter μ (constraint (3.10e) is satisfied) [3].

Hence, the states that a user can be, υ_u, are represented by

$$\upsilon_u = \begin{cases} 1, & \text{if } E'_{u,\mu} < E_{u,\mu}\,, \\ 2, & \text{if } E'_{u,\mu} \geq E_{u,\mu}. \end{cases} \tag{3.13}$$

In terms of reward, r_{υ_u,a_u}, for both states the reward is given as the relative difference between what was achieved and what was requested. The reward is

[3]Since a lower value of latency is preferred, (3.13) will have its signal changed when the latency parameter is considered.

defined as

$$r_{\upsilon_u,a_u} = \frac{E'_{u,\mu} - E_{u,\mu}}{E_{u,\mu}}, \tag{3.14}$$

and represent how far away each user is from being satisfied or how much a user is over satisfied.

For each state-action pair and reward, each agent updates its μ Q-Tables independently for each parameter, resulting in a change of weights for each user. In this case, each Q-Table is represented by a matrix $[a_u \times \upsilon_u]$ and for each time-step of the algorithm, they are updated via (2.1). Lastly, similar to the SC learning, the same two stopping criteria were devised for the user weights learning, in which the algorithm stops either after a fixed number of maximum iterations (M_{uw}) or after its reward did not improve more than a threshold from one iteration to the other.

3.4.3 Proposed Algorithm

Based on the system model and learning phases, an iterative algorithm for the proposed solution can be elaborated, in which the optimization of SCs CREOs and users weights is performed. The proposed solution is distributed, in which SCs update their CREOs independently from other cells and users also update their weights independently from one another. Furthermore, the algorithm is composed of two different parts, the first, *SC learning*, deployed in every SC of the network, performs an optimization of CREOs. The second, *user weights learning*, is deployed in all users devices, and optimizes user weights in order to achieve user-specific cell association.

In terms of the network optimization, each algorithm in every SC needs to have certain parameters initialized, such as backhaul characteristics (load and η parameters, mainly: capacity, latency and resiliency). In addition, SCs CREOs, Q-Tables, and the number of users connected to it are all initialized as zero when the cells are turned on. In terms of users requirements, they could be initialized by different applications, such as whenever an audio/video stream application is open, a higher preference for high throughput and low latency could be requested.

It is envisioned that the SC learning takes place whenever the network detects that its performance is below a threshold, for example, if the total network throughput is below a certain value. As such, whenever this condition is triggered, each SC learns the best CREOs that satisfy the majority of the users connected to them. These offsets depend not only on the state the cell is currently in, and environment conditions, such as shadowing, backhaul load and the number of available RBs, but also on user's requirements and weights. Based on that, each

cell selects the best action, according to what it knows, for each time-slot. In addition, due to the way the problem is formulated and the way that the states are given by the constraints defined in (3.10), the SCs will always be in only one of the possible three states. Hence, for that time-slot, depending on the current SCs states, they will try to find the optimal CREO that maximizes the system total reward, as given by the RL formulation. In other words, the RL optimization problem can be seen as a system that tries to maximize its total reward, by dividing its goal (total cumulative reward) into smaller micro goals (maximize the reward of each iteration). As such, in every iteration, independently of the state a SC is, it will always try to find the best solution for that time slot. In addition, since there is a certain correlation between successive time slots in the network, the SCs keep their Q-Tables between time slots, in order to utilize previous gathered knowledge in order to find better actions in the future.

At the end of this stage, the new CREOs are communicated to the users via the control channel. For the user learning, it is planned that users can change their weights whenever their perceived QoS is below a target. This can happen due to several reasons, such as changes in SCs CREOs, network failures or outages, or network congestion. If a user triggers its learning, the best weights that are assigned to each of its μ parameters are going to be learned. Similar to the learning of SCs, each user evaluates the best actions that it can take based on its current state for that time-slot, which depends on parameters such as the RSRP, the user's position, and the SCs' CREOs. Then, for each parameter, the users choose the best available weight, while keeping μ Q-Tables between time-slots. Similar to the SCs scenario, the Q-Learning of user weights can also be seen as each agent trying to maximize their total cumulative reward (being satisfied with respect to each parameter), but by dividing it into iterations, instead of each agent trying to maximize just one global goal, smaller goals at every iteration are pursued. After the user update its weights, the network association process, according to (3.3) is performed in order to decide if the user stays in the same cell or is handed over to a better more fitting cell.

Because of this iterative process, it is inevitable that ping-pongs occur in the network. However, due to the way the system is modelled, ping-pongs can only occur whenever a cell does not have enough resources to accommodate a user or if the user channel conditions are not good enough, resulting in a poor SINR. However, if any of these conditions are true, the user should be reallocated to a better cell anyway, independently of the proposed algorithm. In addition, because the proposed algorithm only occurs whenever certain thresholds are met, meaning that the network is not operating at its optimal point or that users are

not satisfied, users should also attempt to connect to another cell, resulting in no number of increased connections.

Furthermore, it is envisioned that the weights learned by each individual user are kept in his device and can depend on the type of application being utilized. As such, the proposed solution presents no issues regarding the utilization of different applications. On one hand, regarding mobility management, the proposed algorithm presents the same issues as current solutions for heterogeneous networks, in which user devices in idle mode are continuously ranking potential serving cells. On the other hand, the proposed framework is more robust and can adapt to changes faster, as user devices have the advantage of performing a user-centric selection based on learned weights, while also utilizing previous historical data and gathered knowledge.

Algorithms 2 and 3 show an implementation of the SCs and users learning, respectively, while Figure 3.2 shows a diagram of the overall proposed solution. In Figure 3.2 it can be seen that both users and SC keep monitoring their performance in order to decide when to trigger the proper algorithm. The diagram shows that user 1 (UE1), in active mode, keeps monitoring the network at certain time instants (which can be defined according to application, for example) and when it detects that the performance is below a threshold it triggers Algorithm 3, updating its weights. After that UE1 then changes SC and re-evaluates the network, determining that its condition is back to the desired level. It can also be seen that the SC monitors the network performance and whenever the network conditions are below a threshold it triggers Algorithm 2 resulting in a change of CREOs. These new CREOs are then broadcast to all users, independently if they are in idle or active mode and also of Algorithm 3. Lastly, the diagram also shows what happens if a user is in idle mode (UE2). In this case, when UE2 joins the network, it first performs an initial cell selection, to determine which to camp on and then, after new CREOs are received, it re-evaluates the cell selection procedure to determine if it will remain or HO to a new SC. When UE2 is in idle mode it only needs to reselect cells when new CREOs are broadcast and no user weights optimization is performed in this stage. Only after UE2 has moved from idle to active mode that it starts monitoring the network and performing Algorithm 3, if necessary.

3.4.4 Complexity, Convergence and Overhead Analysis

The proposed solution is analyzed in terms of computational complexity, convergence and signaling complexity to highlight its feasibility and practical implementation.

Algorithm 2: Small Cells Q-Learning

inputs : backhaul conditions, cell load, $\mathbb{E}$, $\mathbb{W}$
output: $\mathbb{O}$

```
for all small cells do
    for each parameter η do
        for all iterations do
            Measure λ_c and θ_μ
            Determine SC current state using (3.11), υ_c
            Choose action: select new CREO value, O_{c,η}
            Determine reward using (3.12), r_{υ_c,a_c}
            Perform action: change SC CREO value
            Measure new λ_c and θ_μ
            Update SC state
            Update Q-Tables according to (2.1)
            if Stopping Criteria is met then
                Stop
            end
        end
    end
end
Return 𝕆
```

Algorithm 3: User Weights Q-Learning

inputs : RSRP, $\mathbb{E}$, $\mathbb{E}'$, $\mathbb{O}$
output: $\mathbb{W}$

```
for all users do
    for each parameter μ do
        for all iterations do
            Measure user (dis)satisfaction
            Determine current user state using (3.13), υ_u
            Choose action: select new weight, W_{u,μ}
            Determine reward using (3.14), r_{υ_u,a_u}
            Perform action: change user weight
            Measure new user (dis)satisfaction
            Update user state
            Update Q-Tables according to (2.1)
            if Stopping Criteria is met then
                Stop
            end
        end
    end
end
Return 𝕎
```

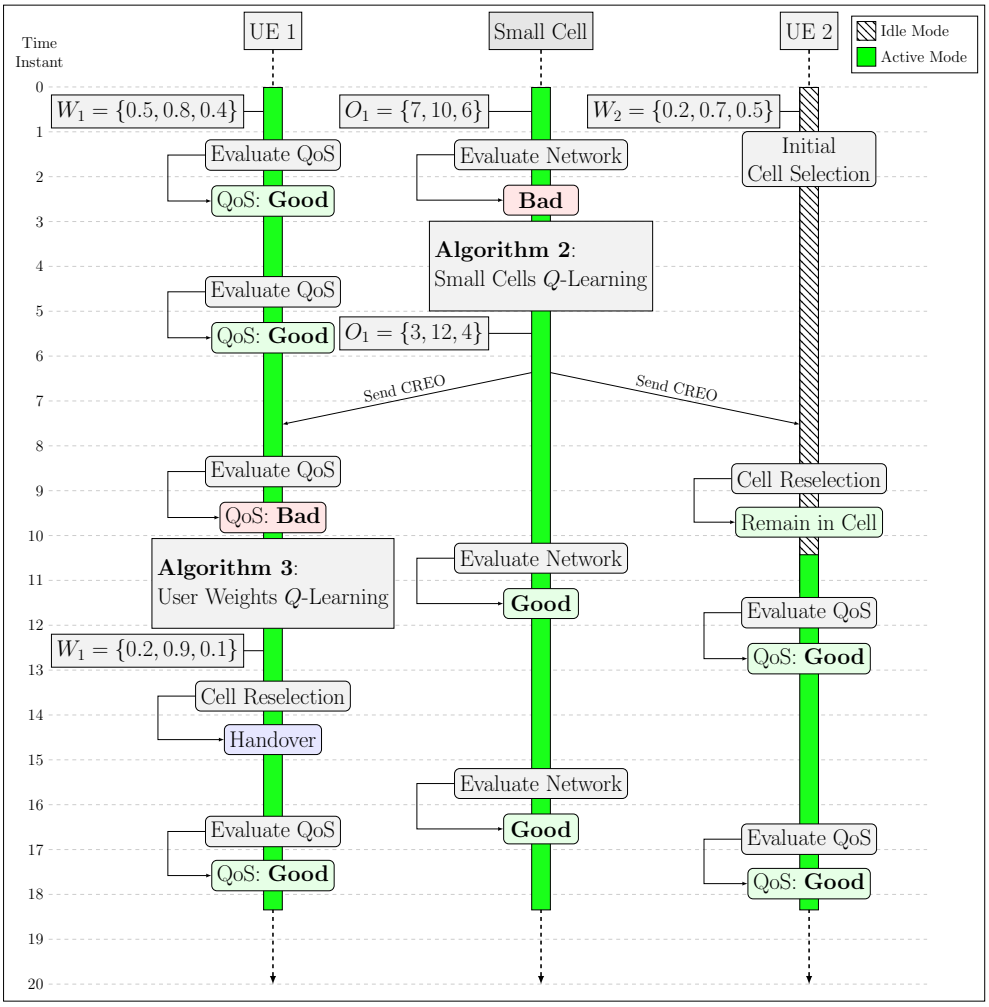


Figure 3.2: Diagram showing how the proposed solution can work. In this diagram, only 2 users are shown for convenience, but it is assumed that more users are connected to the SC. Both users and SC monitor the network and change their parameters if the performance is below a threshold.

Complexity Analysis

It is shown in [203] that for the worst case scenario, the Q-Learning complexity scales linearly with the number of states and actions, assuming a computational complexity of $O(s \cdot a)$, where s denotes the total number of possible states, and a represents the total number of possible actions. For the proposed algorithm, since the η CREOs and μ weights optimizations can run in parallel, the increase in complexity for each SC is given by $O(\upsilon_c \cdot a_c)$, and each user would also require an extra computation of $O(\upsilon_u \cdot a_u)$.

When compared to the fixed CREOs solutions, the proposed solution is slightly more complex, however, this extra complexity at both the network and user sides translate to extra QoS gains (around 10%). When compared to the BS-centric solution, the proposed method adds only an additional level of complexity at the

user side, but as the results show, this increase in complexity is traded-off by gains in user satisfaction.

Convergence Analysis

Q-Learning has already been shown to converge independently of the policy chosen in [10, 50]. As previously mentioned, the RL optimization problem can be seen as a system that tries to maximize its total reward by dividing the problem into smaller micro goals. Hence, from a convergence perspective, it can be said that the algorithm converges at every episode[4] (network snapshot), while also maintaining its Q-Tables in between episodes. In other words, the proposed solution attempts to find, for the current network configuration, the best CREO and weights settings. In addition, although the network changes in between episodes, there is a quite strong correlation between successive time instants, hence the algorithm is able to take future actions based on previous knowledge and maximize its total reward.

Overhead Analysis

The proposed scheme can be implemented in current LTE networks with minimal modifications to the standards. One possible modification may be a small change in the current CREO settings supported by LTE, in which the optimized CREO values of each cell are broadcast using different frequencies and a cell identifier [204]. In this case, the frequency of broadcasting the optimized CREO values would remain unaltered, whereas the frequency in which UEs in idle mode access this information may be changed as it depends on specific implementation. It may also be beneficial that users in idle mode change the frequency in which they access the CREO information, although it is not necessary. As such, the proposed changes would be to associate multiple offsets with every neighboring cell, requiring only $n \cdot \eta \cdot b$ extra overhead, where n represents the number of SCs, η is the number of extra parameters, and b is the number of bits currently used for one offset. One possible alternative to deal with this is to design a system in which the CREOs are broadcast one after the other repetitively in such a way that users are signaled the number of offsets to expect and how often they should get an update. If that is the case, then no additional signaling for broadcasting the CREOs is required.

[4]Whenever there is a notion of time steps in RL algorithms, the agent-environment interaction can be broken into subsequences. These subsequences are called episodes and are composed of repeated interactions between the agent and the environment, until a terminal state, or a stopping criteria has been met [10].

Another source of overhead increase is the need to continuously inform all neighbors of all dynamically optimized CREOs (over the X2 or the S1 interface). This additional overhead has the same cost as before, as $n \cdot \eta \cdot b$. However, despite the increase in overhead, the user-specific scheme is advantageous from a signaling perspective when compared to a user-QoS or backhaul constraint agnostic association policy. The reason is that, in the latter, the probability of a user associating with an unsuitable cell is higher, leading to HOs being triggered to improve user QoS. Hence, by reducing the number of HOs in the network, providing a better user-cell association should be advantageous. Compared to current systems, the proposed scheme increases the signaling proportional to the number of cells, but reduces the overhead in proportion to the number of users, so the cumulative overhead is expected to reduce considerably. Lastly, the user weights optimization does not require that users send their weights to the network, as each user will perform its own optimization, and this optimization can be implementation specific, depending on vendors or applications, and does not require any changes in current standards.

3.5 Simulation Results

3.5.1 Simulation Scenario

In order to provide a proof of concept, an illustrative simulation scenario was set up in MATLAB. For this scenario, a single macro cell, with $m = 3$ sectors was considered, and, on top of each sector, $n = 7$ SCs were overlaid in a random manner. Each SC is considered to have one backhaul link, which can be of one of four types: optical fibre, mmWave, microwave or copper wire. Each backhaul has $\eta = 3$ attributes that define its performance, as seen in Table 3.1, in terms of: capacity, the total data rate that each backhaul is able to support; latency, the delay that users experience if connected to that link[5]; resilience, the reliability of the connection.

In each sector $k = 315$ users are distributed. A third of the users were distributed uniformly and randomly all over the sector, while the other two thirds were uniformly and randomly distributed near the SCs. It is also considered that each user has $\mu = 3$ requirements based on throughput, latency and resilience. In the simulated environment these requirements were generated randomly, assuming that users had an equal probability of requesting either a low or high value

[5]It is assumed that other latencies, such as queuing delay, or the delay caused by different ABS patterns can be dealt with other state-of-the-art algorithms, and that the backhaul latency is the minimum latency that can be achieved, bounded by the fixed link.

Table 3.1: Backhaul Parameters [1, 2]

	Capacity (Mbps)	Latency (ms)	Resilience (%)
Fibre	500	1	99.999
mmWave	500	3	90
Microwave	100	5	99.999
Copper	50	10	99.999
Macro	∞	1	100

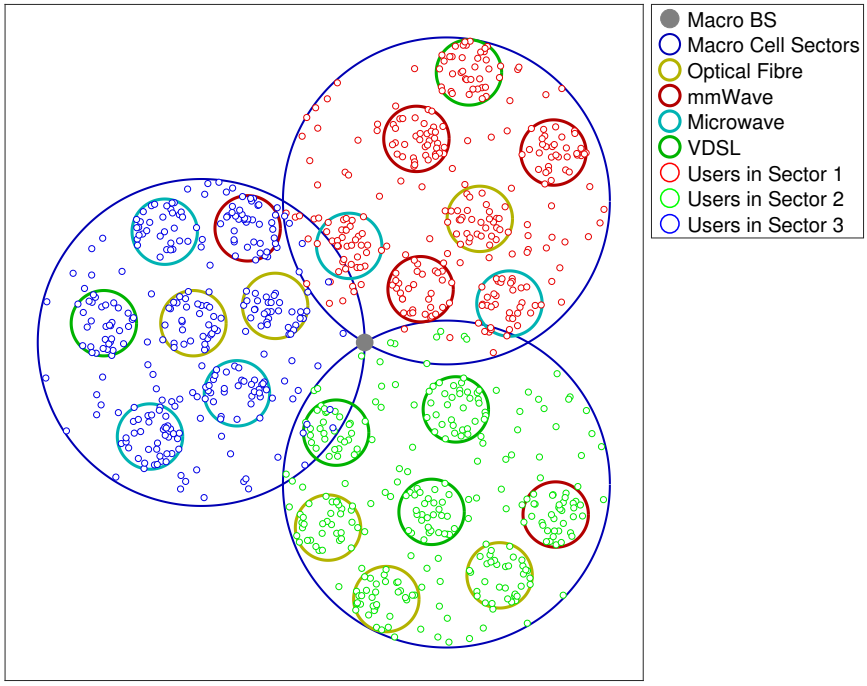


Figure 3.3: Simulation scenario. The macro BS in the centre (in grey) covers an $m = 3$ sectored area (dark blue circles). On each sector, $n = 7$ SCs, with different backhauls (represented by different colors), and $k = 315$ users are randomly distributed, with higher concentration near the SCs.

for each requirement, however, in a real situation, these could be dictated by the application. Figure 3.3 shows one possible configuration of the scenario, in which the macro cell, represented by the grey dot in the centre, covers a three sectored area represented by the dark blue circles. On top of each sector, 7 SCs are randomly positioned, each with a different backhaul connection, and 315 users are overlaid. Table 3.2 shows the simulation parameters, which conform to 3GPP specifications as proposed in [6].

The system is evaluated for a total of ten independent runs, with different starting conditions, such as user requirements and positions, SC locations and backhaul links. At the beginning of each run the η and μ Q-Tables of SCs and users, respectively, are initialized to zero, but as previously mentioned, the corresponding matrices will be maintained in between episodes, being reset only after

another episode begins. Also, other parameters such as channel conditions (fading and shadowing), backhaul loads, and user positions vary from one episode to another. In each run, a total of fifty episodes are performed and the metrics are computed and averaged out. In addition, for the first episode of the algorithm, an allocation process based only on the RSRP is done, so that a real network scenario with users already allocated to the cells of the system can be simulated. During the other episodes of the algorithm, the user-specific solution, based on Q-Learning, is evaluated. The computed metrics are then averaged out, in order to measure the performance of the system and evaluate the robustness of the proposed solution. Moreover, each episode is assumed to be one snapshot of the network, in which network conditions remain static and the SCs and users perform their optimization process over a certain amount of iterations (according to their stopping criteria). For example, in every episode it is assumed that channel and network conditions, such as RAN and backhaul, as well as user mobility remain the same. This is performed for the sake of simulation and in a real system, this optimization would be done in real time. Lastly, for the mmWave backhaul an outage probability is assumed and it is evaluated in every iteration of the algorithm. When an outage occurs, users perceive a very low RSRP from that SC (e.g. -500dBm) and no connections to that cell are allowed in that iteration.

3.5.2 Performance Metrics

The proposed solution is compared to the BS-Centric approach [2] and both 6dB and 12dB fixed CREO. The performance is measured in terms of four metrics: cumulative throughput; total number of users in outage; percentage of unsatisfied users for each parameter; dissatisfaction rate for each parameter. The dissatisfaction is defined as the percentage difference between what was requested and what was allocated, considering only unsatisfied users, as

$$D_{\mu} = \sum_{c=1}^{|\mathbb{C}|} \sum_{u=1}^{|U_c|} \frac{E_{u,\mu} - E'_{u,\mu}}{E_{u,\mu}}, \quad \forall u \in \mathbb{U} | E_{u,\mu} < E'_{u,\mu}, \tag{3.15}$$

where $\mu \in \{T, L, R\}$, T denotes throughput, L corresponds to latency, and R to resiliency. In addition, the penalty incurred in throughput due to ABS is considered in the results. Lastly, the relative gain of the proposed solution with respect to the other methods, G_{us}, is also measured and is calculated as

$$G_{us} = 100 \cdot \frac{F_{\text{us}} - F_{\text{base}}}{F_{\text{base}}}, \tag{3.16}$$

Table 3.2: Simulation Parameters [3–7]

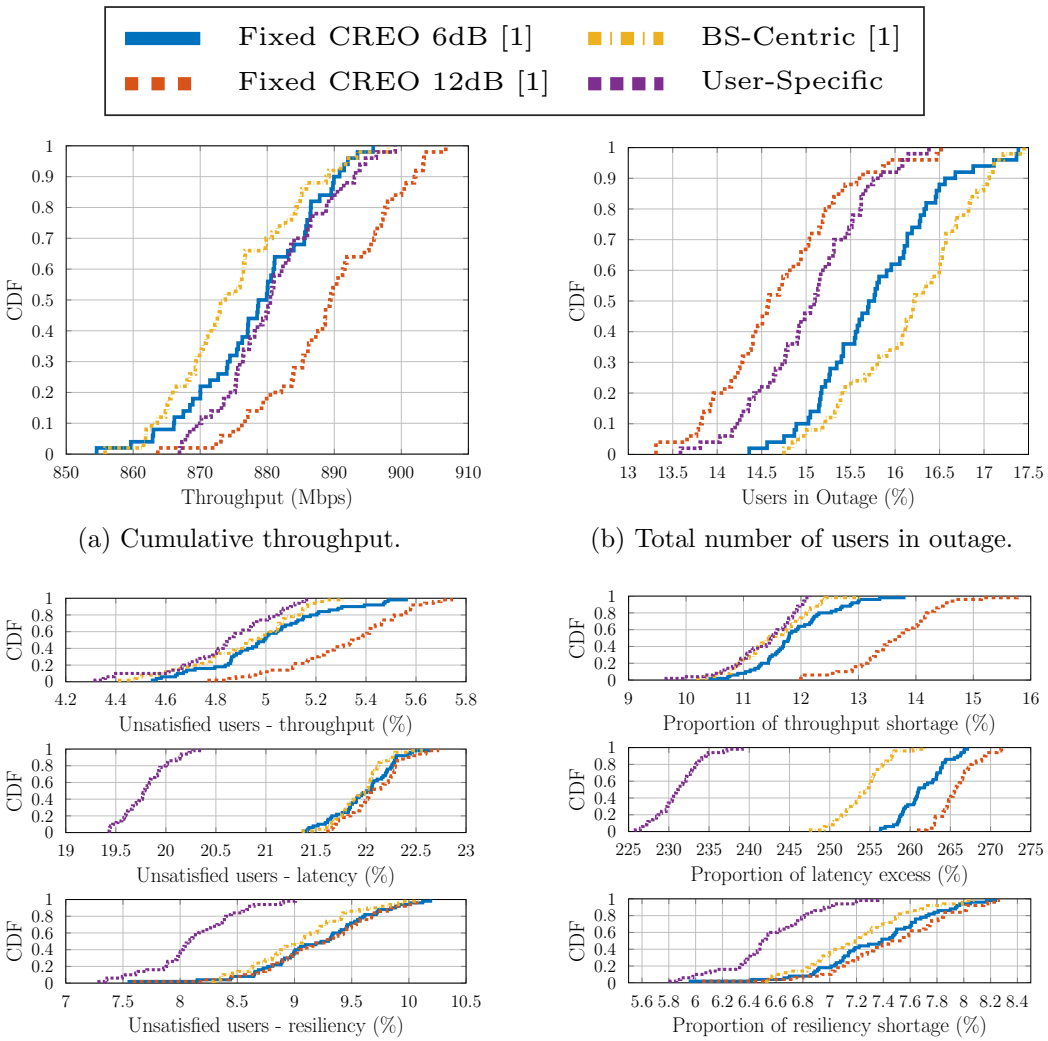
Parameters	**Value**
Number of Sectors (m)	3
SCs per Sector (n)	7
Users per Sector	315
Ratio of Users in SCs	2/3
Sector Radius	250 m
SC Radius	50 m
Macro BS EIRP	20 dBW
SC EIRP	7 dBW
Macro Cell Shadowing	4 dB
SC Shadowing	5 dB
Receiver Noise Figure	7 dB
Penetration Loss	18 dB
RBs per Cell (RB_{max})	50
Backhaul Overhead Factor (ρ)	1.3
Bandwidth of 1 RB (B)	180 kHz
Number of sub-carriers (N_{sc})	600
mmWave Outage	16%
ABS pattern (ζ_{ABS})	40%
Satisfaction Threshold (θ_μ)	0
Throughput req.† (low / high)	0.2 / 1 Mbps
Latency req.† (low / high)	5 / 10 ms
Resiliency req.† (low / high)	90 / 99.999%
Learning Rate (α_c, α_u)	0.5
Discount Factor (ϕ_c, ϕ_u)	0.9
Macro Cell Path Loss	$128.1 + 37.6 \cdot \log_{10}(d)$ dB
SC Path Loss	$140.7 + 36.7 \cdot \log_{10}(d)$ dB
CREO Maximum Value (V_{max})	12 dB
User Weights Maximum Value (G_{max})	1
Total number of episodes	50
Max. iterations ($M_{\mathrm{uw}}/M_{\mathrm{sc}}$)	30 / 50
Reward threshold (r_{th})	10%

†Requirement per RB.

where F_{us} is the performance metric with respect to the proposed user specific solution, and F_{base} is the performance metric with respect to the other baseline solutions.

3.5.3 Numerical Results

Figure 3.4a shows the results for the cumulative throughput of the network. As it can be seen, the cumulative throughput is largest when a fixed CREO is applied, performing better for a 12dB CREO. This works as expected, as by artificially



(a) Cumulative throughput.

(b) Total number of users in outage.

(c) Percentage of unsatisfied users.

(d) Dissatisfaction rate.

Figure 3.4: System performance in terms of total network throughput (a), number of users in outage (b), percentage of unsatisfied users for each parameter (c) and dissatisfaction rates for each parameter (d).

increasing the range of SCs, more users are pushed to the SCs, achieving better reuse of the spectrum (more RBs being available). Hence, the 12dB CREO solution achieves the highest cumulative throughput. When comparing the BS-Centric and user-specific solutions, it can be seen that their performance approach the fixed 6dB CREO, with the user-specific solution slightly outperforming both approaches. This also works as expected, as in some cases it is better to apply large CREOs attracting more users to certain SCs, while in others is best to apply smaller CREOs, making users associate with the macro BS more often. Furthermore, because the reward of the intelligent solutions (BS-Centric and user-specific) is not only composed of the cumulative throughput but also depends on other QoS metrics, such as throughput, latency and resiliency, it is natural that

a trade-off is achieved.

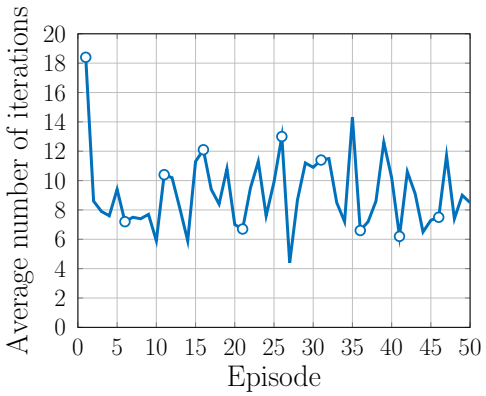
Figure 3.4b presents the total percentage of users in outage for each solution. Also as expected, the 12dB CREO is able to minimize the number of users in outage, as it is able to attract more users, due to the larger artificially extended coverage area. In addition, it can be seen that the user-specific solution lies in between the fixed 6dB and 12dB CREO approaches and that the BS-Centric approach has the worst performance of all. This highlights the gains of the proposed approach, in which tuning user side parameters enables more users to be covered rather than just tuning the CREOs of SCs.

Regarding users satisfaction, Figure 3.4c illustrates the percentage of unsatisfied users in the network with respect to each parameter. It can be seen that fixed CREO solutions do not perform as well as the intelligent solutions, both BS-centric and user-specific. Furthermore, it can be seen that tuning only CREOs of SCs can achieve a better global performance than fixed solutions, but by tuning both CREOs and user weights this optimization can be enhanced. This can be explained by the fact that when both CREOs and weights are considered, together with the proposed constraints, the system tends to deliver what the users have requested, minimizing network resource wastage. This enables more users to be allocated to that SC backhaul, provided that it has enough radio resources available. It can also be seen that tuning both CREOs and weights achieves a better performance with respect to all parameters. Figure 3.4d shows the total proportion of dissatisfaction of users regarding each parameter, which are obtained according to (3.16). As it can be seen, the BS-Centric solution slightly outperform the fixed approaches in all metrics, however in the case of the proposed user-specific approach, the dissatisfaction with respect to all parameters can be mitigated even further.

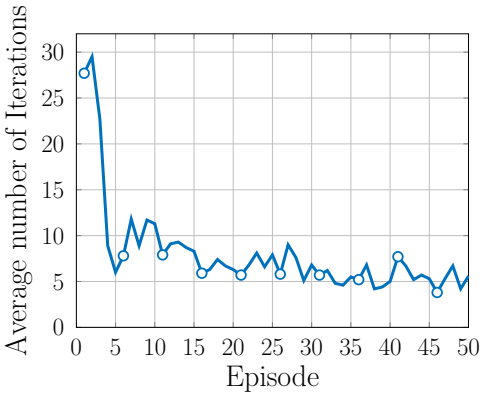
Table 3.3 shows the relative gain of the user-specific solution with respect to other methods. As it can be seen, by optimizing both network and user parameters, the proposed solution is able to reduce the number of unsatisfied users and their dissatisfaction rates by around 10%. Furthermore, when compared with the BS-Centric approach in terms of throughput, it can be seen that both solutions achieve a similar value, indicating that both approaches are able to find near-optimal values for this metric. As it can be seen, Table 3.3 emphasizes that the proposed method is able to better allocate the backhaul resources, reducing the number of unsatisfied users as well as their dissatisfaction rates. By delivering for each user only what is requested, $E'_{u,\mu} \rightarrow E_{u,\mu}$, the amount of resources allocated to over satisfied users is reduced, freeing backhaul resources and reducing the number of unsatisfied users and their dissatisfaction rates. However, this comes

Table 3.3: Relative gain (in %) of the User Specific solution

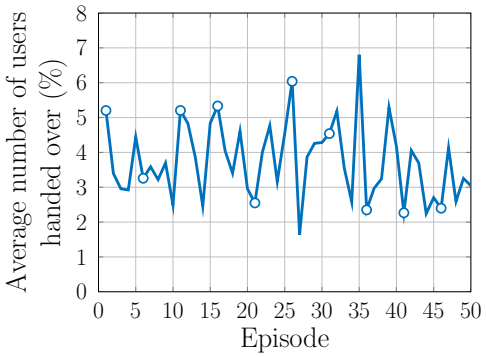
	Relative Gain (%)							
	Total Throughput	**Users in Outage**	**Unsatisfied Users with respect to**			**Dissatisfaction Rate with respect to**		
			Throughput (T)	**Latency** (L)	**Resiliency** (R)	**Throughput** (T)	**Latency** (L)	**Resiliency** (R)
Fixed CREO 6dB [1]	0.26	4.91	3.39	11.03	13.32	4.53	13.17	12.40
Fixed CREO 12dB [1]	-0.95	-2.38	10.00	11.56	13.62	20.12	14.98	14.16
BS-Centric [1]	0.66	7.52	1.61	10.86	11.52	1.30	9.59	10.03



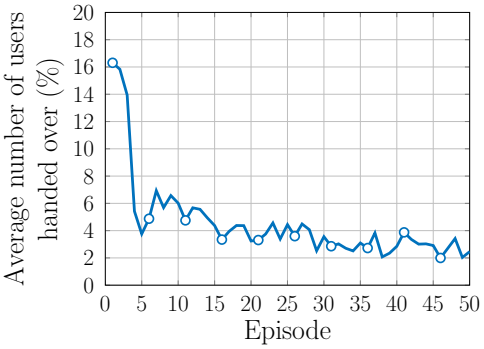
(a) Average number of iterations (SC Learning) per episode.



(b) Average number of iterations (User Weights Learning) per episode.



(c) Average number of users handed over, in percentage, per episode (SC Learning).



(d) Average number of users handed over, in percentage, per episode (User Weights Learning).

Figure 3.5: Analysis of the convergence properties of the proposed algorithm.

at a slightly expense in terms of cumulative throughput and number of users in outage (when compared to the fixed 12dB CREO solution).

Lastly, Table 3.4 shows how users associated to the macro cell and SCs in and out of the CREO regions contribute to the total of unsatisfied users and dissatisfaction rates. As it can be seen, the proposed solution is able to achieve the minimum dissatisfaction amongst CREO users, at the expense of a higher dissatisfaction rate of users connected to the SCs. Also, the user-specific solution associates the second most amount of users to SCs (when accounting both SC and CREO regions), only behind the 12dB approach. However, the user-specific solution associates more users to the macro cell than the fixed 12dB solution. This highlights the objective of the proposed solution, in which depending on the combination of CREOs, user weights and requirements, users are redirected to the most fitting cell, minimizing network resource wastage.

Regarding the algorithm convergence, Figs. 3.5a and 3.5b show the average number of iterations of SCs learning and user learning per episode, respectively.

Table 3.4: Contribution of different users to total number of unsatisfied users and dissatisfaction rates

		Contribution in (%) to total amount of unsatisfied users			**Contribution in (%) to total dissatisfaction**			**Average of associated users**
		Throughput	Latency	Resiliency	Throughput	Latency	Resiliency	-
CREO 6dB	Macro Cell	17.24	0	0	7.00	0	0	150
	Small Cell	48.48	89.19	89.62	86.04	98.58	98.77	583
	CREO Region	34.28	10.81	10.38	6.96	1.41	1.23	63
CREO 12dB	Macro Cell	9.98	0	0	4.77	0	0	134
	Small Cell	32.25	84.70	84.76	63.51	96.55	96.71	566
	CREO Region	57.77	15.30	15.24	31.72	3.45	3.29	107
BS-Centric	Macro Cell	19.95	0	0	8.39	0	0	150
	Small Cell	48.58	89.55	90.98	84.22	98.46	98.91	580
	CREO Region	31.47	10.45	9.02	7.38	1.54	1.09	62
User-Specific	Macro Cell	18.05	0	0	6.28	0	0	150
	Small Cell	55.92	90.40	91.02	89.46	98.80	99.05	593
	CREO Region	26.03	9.60	8.98	4.25	1.20	0.95	59

As it can be seen and as expected, in the beginning, as both algorithms do not know enough about the environment, they start by performing plenty of iterations in order to find the optimal network and user settings. However, as the number of episode increases, this number decreases and both solutions converge to around 10 and 5 iterations in case of SC learning and user weights learning, respectively. Moreover, it can be seen that the optimization of user weights is more stable because they operate after the SCs have optimized their CREOs. On the other hand, the optimization of CREOs is slightly more unstable, although it still converges, due to network changes and user mobility, which varies from one episode to the other. It can also be seen that the proposed solution converges rather fast, as both algorithms converge after around 5 episodes.

Figures 3.5c and 3.5d show the average number of users handed over per episode after performing each algorithm (in percentage). As it can be seen, when SC learning is performed an average and constant number of 4% of total users is reallocated every time, while when users learn their weights, this number starts relatively high at around 16% and then converges, after around 20 episodes, to around 3%. This not only shows the convergence of the proposed methods, but also further emphasizes that by only tuning CREOs a constant rate of users is handed over to SCs, while by tuning both CREOs and user weights the algorithm can learn which users to HO and only change the association of the users that it needs to.

3.6 Summary

In order to achieve the requirements of future mobile networks, such as the ever increasing user demands and also to enable a wide range of applications, it is clear that intelligent and robust solutions need to be deployed. With that in mind, new paradigms of user-cell association need to be considered, in which the end-to-end connectivity is contemplated, instead of current radio interface based solutions. In addition, solutions must also optimize not only parameters of the network, but also user parameters, to achieve user-specific cell association.

As such, a RL approach, in which both SC CREOs and user weights are optimized, using Q-Learning is proposed. Results show that the proposed method outperforms fixed CREO solutions and another BS-centric approach. Results also demonstrate the importance of tuning both network and user side parameters, as this enables the proposed algorithm to allocate only enough for each user in order for it to be satisfied, while also allowing more backhaul resources to be shared among other users. Thus, by optimizing both network and user parameters a re-

duction of around 10% in the total number of unsatisfied users and dissatisfaction rates could be achieved, highlighting the potential that intelligence can bring to future networks in terms of user satisfaction and network resource optimisation.

Chapter 4

Intelligent UAVs for Emergency Mobile Networks

4.1 Introduction

Although sporadic, natural large-scale disasters, such as earthquakes, hurricanes, and tsunamis produce a profound impact in human society, not only in terms of the infrastructure that is destroyed, but most importantly, in terms of human lives that are lost. Whenever a disaster occurs, it is crucial that search and rescue teams are deployed in a very quick and effective manner, as the first 48 to 72 hours after a disaster, also known as the *golden hours*, are the most critical [205, 206]. However, one major problem that arises during this period is the lack of communication infrastructure, as most of the existing network can be destroyed during a disaster, degrading the rescue team's effectiveness and ability to find isolated people. Hence, in order to overcome a disaster situation, it is also vital that an emergency communication network (ECN) is deployed as fast as possible, so that communication can be restored quickly, preventing additional casualties.

In general, ECNs must be extremely adaptable, flexible, and intelligent in order to adjust themselves to the environment and situations that they can be inserted in. Hence, conventional solutions that are found today in wireless mobile networks may not be applicable, as deploying a completely new network from the very beginning can take several days. This can occur either due to the conditions in which the environment is, such as a city being completely devastated, limiting the accessibility to certain areas, and by consequence, limiting network coverage, or due to the complex process of configuring and setting up all new BSs parameters. Thus, more robust solutions, involving networks that are capable of self organization and that can be deployed quickly and effectively to the exact area

where coverage is needed should be designed. In this sense, algorithms that can adapt themselves, such as artificial intelligence and machine learning, should be deployed [8, 205–207], to enable a fully autonomous network.

One possible solution for a rapid deployment of an ECN that can provide the desired flexibility, quickness and intelligence is the utilization of drones with wireless capabilities [205, 206, 208–210]. These UAVs, also known as UAV base stations (UAV-BSs), would be equipped with a small BS, and would serve as radio access points in the network. In addition, since after a disaster reaching certain locations of the affected area can be problematic, due to debris blockage or flooding, for example, positioning conventional BSs can be a rather difficult task for network operators. Thus, because of their flexibility and mobility, drones can perform a crucial role in emergency situations, by flying to the affected area and providing service exactly where it is needed.

Based on this, a positioning algorithm for UAV-BSs in an emergency situation is proposed. The solution aims at finding the best position of multiple drones in a scenario where the old mobile network infrastructure was completely destroyed and users are in need of coverage. The UAV-BSs are considered to have limited resources in both radio access network (RAN) and backhaul. In addition, users could have different requirements in terms of throughput and different mobility characteristics, depending if the user is from a rescue team or a regular user. As such, a distributed algorithm, based on RL, more specifically Q-Learning, is proposed, so that the drones can explore the affected area and find the best possible position. This way, the main objective, to maximize total network coverage (or minimize the number of users in outage), can be achieved. The proposed solution is compared to different positioning strategies, such as deploying the drones in fixed random positions, fixed around a circle centred in the middle of the area at evenly spread angles, and deploying the drones in the locations of hot spots of the previous destroyed network and the results show that the intelligent Q-Learning solution outperforms all of them in all considered metrics.

4.1.1 Related Work

Aerial platforms, such as drones, are expected to have an important role in the next generation of mobile networks. Because of their flexibility, adaptability and mobility capabilities, these platforms can be deployed in a wide range of situations, ranging from providing extra coverage and capacity whenever a big event takes place, supplying the necessary communication infrastructure in case of an emergency, or bringing service in rural and isolated areas, to name a few. Because of these reasons, the deployment of drones in mobile communication

networks has seen an increased attention recently [205, 207–218]. In addition, the deployment of machine learning solutions in mobile networks, more specifically self organizing mobile networks, has also seen an increase in recent years and research groups all over the world are developing intelligent solutions in order to tackle the various challenges of mobile networks [8].

Erdelj et al., in [205], present a survey of the advances in drone technology focused on wireless sensor networks and disaster management. The survey divides a disaster into three main stages, mainly: pre-disaster preparedness; disaster assessment; and disaster response and recovery; and presents drone applications and challenges for each one of them. In [211], the authors show key aspects of the design and implementation of future aerial communication networks, however instead of focusing on small drones, the authors focus on tethered balloon platforms.

Other works, such as [212–214], attempt to find the best position of UAV-BSs analytically. In [212], for example, the authors attempt to find the best position for low altitude platforms (LAPs) in order to maximize their coverage range. The authors develop an analytical solution to determine the best altitude of a LAP and end up concluding that the optimum altitude is strongly dependent on the environment. Mozaffari et al., in [213], derive the optimal altitude of UAV-BSs which gives the maximum coverage, while minimizing the transmit power. The system is investigated in two different scenarios, one considering interference between drones and another being interference-free. Results showed that, when interference is considered, there is an optimal separation distance between drones in order to maximize the network coverage. In [214], Alzenad et al. present an optimal placement algorithm for UAV-BSs that maximize the coverage while minimizing the transmit power of the drones. In addition, the authors decouple the problem in two, considering the placement of the drones as two separate problems in both horizontal and vertical dimensions. Results show that their system is able to save a significant amount of power, while also increasing the number of covered users.

Kalantari et al., in [207], propose to find the best position of UAV-BSs, but instead of determining it analytically, they utilize a particle swarm optimization (PSO). Their results show that PSO is capable of adapting to different scenarios and that the drones are able to find by themselves the best positions in order to maximize the number of users being covered. Ahmadi et al., in [208], propose a novel mobile network architecture, considering drones as a core part of the network. Their work formulates the optimum placement of drones, while also presenting some challenges and future research directions. Also regarding the

positioning of drones, Merwaday et. al. show in [215] that, in an emergency scenario, finding the optimal position for temporary UAV-BSs via exploiting the mobility of the drones yields improvements in network throughput and spectral efficiency.

Another work by Kalantari et al., in [216], investigates the usage of flying base stations considering different types of backhaul links. The authors introduce two different approaches, mainly a network-centric approach and a user-centric approach, and determine the best three dimension (3D) position of UAV-BSs. Their results show that the network-centric approach is able to maximize the number of covered users, and that the user-centric solution maximizes user throughput. Another paper which considers backhaul limitations is the work in [214], by Alzenad et al., wherein the authors study the feasibility of a novel backhaul framework considering aerial platforms and free space optics point-to-point links. Their results demonstrate that this type of backhaul is capable of delivering higher data rates than others, but it is also very sensitive to the environment, including clouds and fog. In [210], the authors consider the utilization of drones as a complementary approach to future terrestrial mobile networks. The authors present some design opportunities and challenges, and also develop a case study on the positioning of UAV-BSs.

Mozaffari et al., in [209], present the deployment of a drone network on top of an already existing device-to-device network. The authors evaluate the system in two different scenarios, considering static and mobile drones. The authors derive the outage and coverage probabilities for each case and show that the mobile strategy performs better than the static one in terms of coverage and energy efficiency. Azari et al., in [217] propose a framework for the analysis and optimization of air to ground systems considering altitude and cooperation diversity. The authors consider drones as relays and develop analytical solutions for the drones height in order to maximize its reliability and coverage range. Lastly, Shah et al., in [218] propose a new solution to the problem of user cell association considering flying BSs with backhaul constraints. The authors present a distributed solution based on a greedy search algorithm and show that the proposed approach has better results than other baseline approaches and it is less computational complex.

Despite some works covering the deployment of drones in emergency situations [205, 211], other works covering the deployment of drones with backhaul limitations [214, 216, 218], and others considering the positioning of aerial platforms [207–210, 212–214, 217], only [207] proposes an intelligent solution in order to determine the best position of UAV-BSs. Also, as it can be seen from the

reviewed literature, most studies address the drone positioning problem analytically, through the development of closed-form equations. These methods, although important, require several assumptions, such as the knowledge about how many users are in the network and their positions. In addition, most of these works also do not take into account user mobility and perform the drone placement optimization for a specific, static scenario. Hence, these types of solutions might not be suitable for real situations, in which the environment is constantly changing, users can move at different speeds and even network parameters, such as cell load and backhaul conditions, can change as well.

In addition, as previously mentioned, the only work that proposes an intelligent solution to the problem of drone positioning optimization is the work of Kalantari et al., in [207]. However, the proposed work utilizes a PSO algorithm, which can be viewed as a branch of GAs or heuristic methods (in contrast to GAs, PSO does not perform selection in between generations) [219,220]. Although able to solve the proposed problems in a simulated environment, solutions such as GA, heuristics, and PSO, due to their inherit nature of having to search for the best possible solution amongst a family of available ones, are not suitable for applications that require continuous interaction between the system and its environment. This occurs because any change in the initial original set of solutions, would require the whole computation to be performed again. For instance, PSO is not able to perform an online optimization of the problem.

As the authors show in [207], the approach is tested in two fixed scenarios, without considering user mobility. Because PSO performs an offline computation, this solution is also not capable of adapting itself to real time changes in the network. For example, if mobility was taken into account, the proposed PSO algorithm would have to run again, every time a user would move, in order to determine the best new solution for this new network configuration, resulting in an impractical system. Additionally, due to the vast search space that the PSO solution has to evaluate, a centralized unit would be required in order to perform all the required computations and determine the best configuration. Again, in real systems this is not practical, as this would result in an increase in communication signalling between the centralized unit and the drones, as well as the need of synchronization. Lastly, due to the heuristic nature of PSO, this approach would also not be scalable as well as computationally efficient due to the vast search space that it must compute in order to find the best possible configuration. In a real environment, for example, in which network conditions and user positions change frequently, PSO would not be able to cope with these changes, becoming an impractical solution in real scenarios.

Based on the issues mentioned above, it is clear that the development of a novel solution that is capable of adapting itself online and that is also able to analyse the environment and determine the best possible actions to be taken is needed. Based on that, RL algorithms can be a suitable approach since, independently of the environment they are inserted in, they can explore the possibilities and determine the best actions to be taken.

4.1.2 Objectives and Contributions

The main objective is to provide a distributed and intelligent solution to the problem of positioning multiple UAV-BSs in order to maximize the number of covered users in an emergency situation. As previously mentioned, this problem is of high importance in emergency situations, since the fastest a communication network can be established, more human lives can be saved. This optimization problem also poses a difficult challenge, due to the varying conditions of the environment, such as users moving with different speeds, users having different requirements and the UAV-BSs being limited in both RAN and backhaul resources. In addition, as the deployment of conventional BSs might not be possible, either due to the infrastructure being destroyed or parts of a city not being accessible, the deployment of drones is considered to be a good solution in case of emergency situations.

In order to provide a solution to the problem of drone 3D placement, the development and evaluation of an adaptable and flexible solution based on RL using Q-Learning is proposed. The developed algorithm is shown to be robust enough in order to adapt itself to different network conditions, such as the position of other UAV-BSs, interference between UAV-BSs, user movements and requirements. To the best of the author's knowledge, there are no other works that consider the optimization of drone positioning using RL in an emergency communication scenario, whilst also considering user mobility, user requirements and network constraints.

The main novelty of this work can be described as the development of an intelligent solution based on RL, in order to tackle the problem of user coverage in an emergency situation. As discussed before, most of the current state-of-the-art solutions do not provide the needed flexibility or adaptability in order to cope with a changing environment, hence the development of an intelligent solution that is capable of providing coverage exactly where and when needed is essential. The proposed solution utilizes Q-Learning in every UAV-BS, which will attempt to find the best position in the environment so that the global reward, given by the total number of users covered, is maximized. Results show that the

proposed solution achieves better levels of outage and user satisfaction than fixing the drones either in random positions, in a circular manner, or in the location of the previous hot spots.

4.2 System Model

The problem of maximizing coverage in an emergency situation scenario, via the deployment of a temporary network is considered. This network is composed of a truck BS, along with movable UAV-BSs. The objective of the proposed system is to find the best possible positions of UAV-BSs, given that users with different requirements and mobility characteristics are distributed in the scenario and that both the truck BS and the UAV-BSs have limited resources in terms of RAN and backhaul.

4.2.1 Urban Model

The International Telecommunication Union-Radio (ITU-R) defines in [221] three parameters to characterize any urban environment, which are defined as:

- α, the ratio of build-up land area to the total land area;
- β, the average number of buildings per square kilometre;
- γ, scale parameter for the heights of the buildings.

Following [221], the urban scenario in this paper is modelled considering these parameters, and a building disposition following a Manhattan grid layout, as in Figure 4.1, in which squares of a given width (W) are separated by a distance (S) [222]. The height of the squares is obtained following a Rayleigh distribution with scale parameter γ, whereas W and S are assumed equal across all buildings and are determined via [223]

$$W = 1000 \cdot \left(\frac{\alpha}{\beta}\right)^{\frac{1}{2}}, \tag{4.1}$$

and

$$S = \frac{1000}{\sqrt{\beta}} - W. \tag{4.2}$$

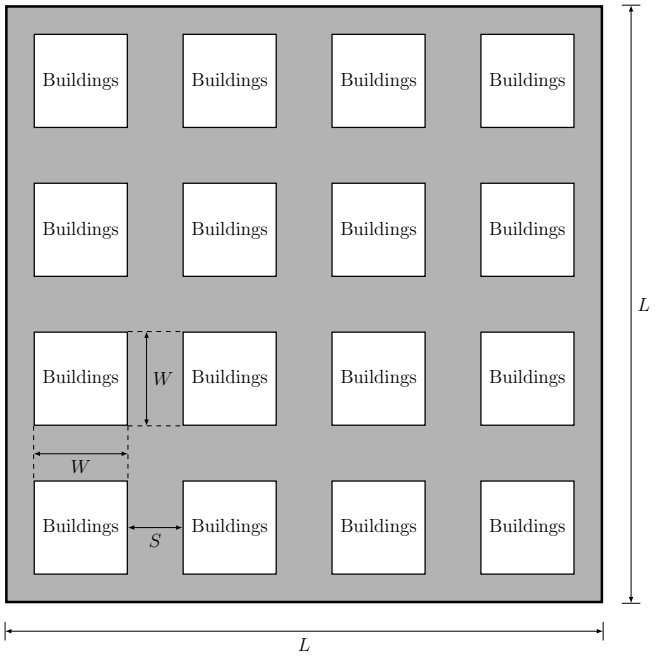


Figure 4.1: Manhattan grid urban layout.

4.2.2 Users

The scenario considers a total of N_{u} users spread across an L by L square area. A portion of the users is assumed to be concentrated near hot spots randomly distributed around the hot spot coverage area, while the rest of the users is uniformly randomly distributed in the square area.

In addition, two types of users are considered: either users that belong to an emergency team, or regular users. Regardless of their type, the most important requisite for all of them is to be connected (to have coverage). Aside from that, rescue team users can have different throughput requirements, depending on their needs, while normal users are all assumed to have low throughput requirements. Furthermore, different mobility levels are assumed between users. Regular users, for example, are considered to have low mobility, whereas rescue team users are assumed to have higher mobility.

4.2.3 Temporary Network

The scenario considered assumes that there is a fully functional network in a certain location, but due to a natural disaster, it has been completely destroyed. In this particular case, however, it is assumed that either part of the original backhaul of the previous network is accessible or that a backhaul link could be deployed after the emergency happened, and could be used by network operators in order to establish a connection to their servers. In other situations, in which

the original backhaul links of the previous network are also destroyed, a solution can be developed first, to deal with the backhaul connections issue, and then the proposed solution, involving the deployment of drones, can be installed.

Truck Base Station

In order for operators to access the original backhaul of the network, a fully equipped truck with a macro BS powered either by fuel or solar energy is considered, such as in [224]. This truck could be placed in a position similar to where the original BS was positioned and, by connecting to the available backhaul link, it can enable the connection between the truck BS and the network operator.

The path loss from the macro BS to the users follows the Okumura-Hata model, which is applicable for frequencies in the range of 150MHz to 1920MHz, but can be extrapolated to frequencies up to 3GHz [202]. The macro BS path loss is defined as

$$\begin{aligned} \mathrm{PL_m} = 69.55 + 26.16 \cdot \log_{10}(f_c) - 13.82 \cdot \log_{10}(h_\mathrm{B}) - C_\mathrm{H} + \\ + [44.9 - 6.55 \cdot \log_{10}(h_\mathrm{B})] \log_{10}(d_\mathrm{m}), \end{aligned} \quad (4.3)$$

where f_c is the carrier frequency, h_B is the height of the macro BS and C_H is the antenna height correction factor, which is given by

$$C_\mathrm{H} = 0.8 + (\log_{10}(f_c) - 0.7) \cdot h_\mathrm{u} - 1.56 \cdot \log_{10}(f_c), \quad (4.4)$$

where h_u is the height of users' mobile devices.

UAV Base Stations

Additionally to the truck BS, it is also considered that UAV-BSs are deployed in the network in order to provide the additional needed coverage. Each drone is considered to have a dedicated out of band backhaul link, composed of a microwave link, which is able to contact the truck BS in order to connect to the network operator. The traffic from the drones are routed to the truck BS and then to the network operator, as the truck BS is the only one with a direct backhaul connection to the operator. Also, it is assumed that the drones use a dedicated spectrum slice of their band to perform this connection to the macro BS. Furthermore, since the drones are flying at relatively high altitudes and the antenna from the macro cell is positioned higher than nearby buildings, it is assumed that the link between drones and macro BS has very large capacity, similarly to what is considered in [215]. Moreover, as recently shown in [225], the noise and

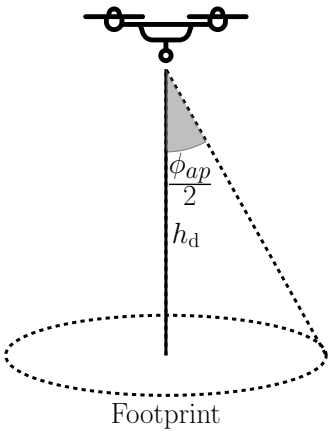


Figure 4.2: UAV-BS flying at a height, h_{d}, and with an antenna with aperture angle of ϕ_{ap}.

LoS characteristics of the channel affect the link between ground BS and drones far less than when considering the communication between the ground BS and a ground user, further supporting this assumption.

It is also considered that the antenna of each UAV-BS has strong directivity, such that its major lobe[1] (defined as the region in which the antenna gain is the highest) has an aperture angle of ϕ_{ap}, as in Figure 4.2. Therefore, the signal irradiated from its antenna is considered highly attenuated outside its coverage radius ρ defined by

$$\rho = h_{\mathrm{d}} \cdot \tan\left(\frac{\phi_{ap}}{2}\right), \tag{4.5}$$

where h_{d} is the drone flight altitude.

The path loss between UAV-BSs and users ($\mathrm{PL_d}$) follows the model presented by Al-Hourani et al., in [223], wherein $\mathrm{PL_d}$ is split between a free-space path loss and an additional loss, which depends on whether there is LoS between the drone and the user or not [213]

$$\mathrm{PL_d} = 20\log_{10}\left(\frac{4\pi f_{\mathrm{c}} d_{\mathrm{d}}}{c}\right) + \xi, \tag{4.6}$$

where d_{d} is the distance between drones and users, c is the speed of light and ξ is the additional loss which assumes different values for LOS or non-line-of-sight (NLOS) links. Figure 4.3 presents the scenario considered in this paper, with a UAV-BS covering regular users and rescue team in a post-disaster urban scenario.

[1]In practical antennas [226], there is one major lobe, which concentrates the majority of its power and can be defined using 2 angles named elevation and horizontal angle. It is assumed in this work that both have the same value, ϕ_{ap}.

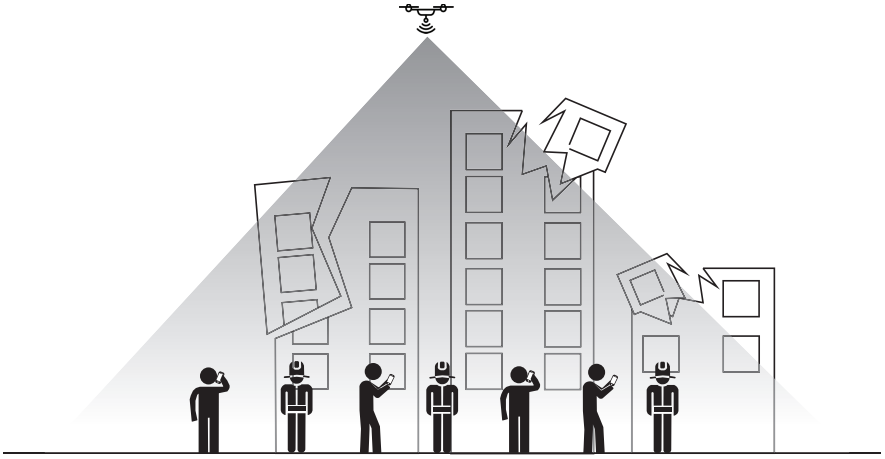


Figure 4.3: Considered scenario. A UAV-BS providing coverage to a certain amount of users, both regular and rescue team users, in an emergency situation.

4.2.4 User Allocation

Considering the set of all users $\mathbb{U} = 1, 2, \ldots, N_{\mathrm{u}}$ and the set of all BSs $\mathbb{B} = 1, 2, \ldots, N_{\mathrm{b}}$, with N_{b} as the total number of BSs, the received signal power, in dBW, for user $i \in \mathbb{U}$, from BS $j \in \mathbb{B}$, is denoted by $\mathrm{RSRP}_{i,j}$ and can be computed as

$$\mathrm{RSRP_{i,j}} = \begin{cases} \mathrm{EIRP_m} - \mathrm{PL_m}, & \text{if } j \text{ is a macro BS,} \\ \mathrm{EIRP_d} - \mathrm{PL_d}, & \text{if } j \text{ is a UAV-BS,} \end{cases} \tag{4.7}$$

where EIRP represents the transmitted power combined with its antenna gain, in dB. According to that, $\mathrm{EIRP_m}$ and $\mathrm{EIRP_d}$ represent the total transmit power from the macro BS and the UAV-BSs, respectively. $\mathrm{PL_m}$ represents the path loss between the macro BS and users, calculated according to (4.3), and $\mathrm{PL_d}$ is the path loss between drones and users as given in (4.6).

Next, the SINR, for a BS/user pair, $\mathrm{SINR}_{i,j}$, is given by

$$\mathrm{SINR}_{i,j} = \frac{\mathrm{RSRP}_{i,j}}{N + \sum\limits_{k=1,k\neq j}^{N_{\mathrm{b}}} \mathrm{RSRP}_{i,k}} \quad , \tag{4.8}$$

where N is the additive white Gaussian noise (AWGN) power and the RSRPs are expressed in linear form.

The throughput T for a user, is determined following Shannon's channel capacity formula, given by (3.5) [227]. As in [2], the amount of throughput that the user consumes from the backhaul is considered to be 30% higher than its actual throughput, because of overhead signals. In addition, if the sum of the backhaul throughput of any BS exceeds its capacity, i.e. if the backhaul is overloaded,

the throughput of all users connected to that cell is reduced equally by taking the amount of overloaded capacity and dividing it by the total number of users connected to that cell. This guarantees that the total throughput of the cell does not exceed its capacity, while also penalizing cells having its backhaul overloaded.

Lastly, users are then allocated to the best BS according to their SINR. If the SINR of a user is above a certain threshold, and the BS has enough space in its RAN, then a user is allocated to that BS in that time slot. However, if that BS has no resource blocks available or the user SINR is too low (below the connection threshold), the next BSs are tried, in order of highest SINR. After all BSs are tried, if a user is still unable to be associated with a BS, the user is considered to be out of coverage (in outage) for that time slot.

4.3 Proposed Solution

The goal of this work is to propose and evaluate an intelligent algorithm, based on RL, to find the best positions of multiple UAV-BSs, which maximize the number of served users in an urban area where a disaster has occurred.

4.3.1 Positioning Algorithm

The proposed algorithm is based on Q-Learning and a distributed approach is assumed. Each UAV-BS is considered an agent, and the disaster scenario (composed of the buildings, user distribution and movement, and macro cell location) is the unknown environment that the agents are inserted in. The states of the UAV-BSs are defined as their three dimensional position in the environment, and each drone can take any of seven possible actions, namely: move up, down, left, right, forward, backward, or not move at all. In addition, the drones follow an ϵ-greedy policy [10] to choose their actions, with a decaying ϵ, depending on the number of iterations.

Since the main target of the proposed solution is to maximize the amount of covered users, the reward experienced by each drone is the total of users allocated by the system. The total number of users allocated is chosen as a reward metric, instead of considering the number of users allocated by each drone, so that drones would have a better incentive to allocate more users instead of each drone being greedy and trying to maximize their own reward. If the individual number of users allocated per UAV-BS was used as reward, this could result in a drone allocating users from other drones, so that its reward would be maximized at the expense of the reward of the others, leading to a sub-optimal (or local) optimization. Thus,

a reward that represented a global metric, in this case the total number of users allocated, was chosen. A summary of the Q-Learning parameters is provided next and Algorithm 4 summarizes the proposed solution.

Agents

Agents correspond to the UAV-BSs. Each agent has a different action-value function, which describes the value of a drone taking an action for every given state.

States

A state s is the three dimensional position of a UAV-BS in space. In addition, for the considered model, the space is divided into a grid, such that the set of possible states a drone can be in is finite.

Actions

Each drone can take seven possible actions. These actions are: moving one step in any direction (up, down, left, right, forward, backward), or to stay still. Furthermore, if an action would take the UAV-BS out of the grid, it has the same effect as staying still.

Reward

Since the goal is to maximize the number of served users, the reward, r, is given by the total number of allocated users. For that to be possible, the UAV-BSs and the macro cell are assumed to share the information of the number of allocated users with each other, via the connection with the macro BS. More formally, the reward of the system can be defined as:

$$r = \sum_{j=1}^{N_{\mathrm{b}}} U_j, \tag{4.9}$$

where U_j is the number of users allocated to BS j.

Update strategy

Each UAV-BS updates its action-value function according to (2.1) The UAV-BSs also keep track of the state which yielded the best reward.

Initialization

At the start, all the UAV-BSs are positioned at random locations and with their action-value functions set to zero in all entries.

Stopping criteria

The stopping criteria is based on three conditions. The drones have moved for a maximum number of iterations Max_{it}, the value of the reward has not improved in a certain number of iterations $\text{Max}_{\text{it,r}}$, or the drone has used all its resource blocks and has explored for a minimum number of iterations Min_{it}. When one of them is met, the UAV-BS moves to the state which yielded the best reward and stops until the next episode.

Episode

In the context of the proposed problem, an episode can be defined as a snapshot of the environment, or in this case, the emergency communication network. In each episode, the UAV-BSs take actions based on their current state and evaluate their reward. This process is repeated for a certain number of iterations until one of the three stopping criteria is met.

In addition, during each episode of the network, the users are considered to be static, so that the drones can evaluate and determine the best actions for that snapshot of the network. After all drones have moved and found the best possible positions for that episode, the current episode ends, users eventually move according to their mobility levels and a new episode begins. Furthermore, since there is a high correlation between episodes, whenever a new episode begins, the UAV-BSs start at the previous position of the last episode. Also, the UAV-BSs keep their Q-Tables between episodes, in order to explore the previous knowledge gathered from previous episodes of the network.

In a real scenario, however, as there is no notion of episodes, the UAV-BSs could perform the position optimization every certain time intervals, for example, whenever the global reward of the system is below a certain threshold. By analysing the network at certain time slots, and moving according to the user positions of that time slot, the UAV-BSs would still be able to find on the fly an optimal solution. In addition, this interval (the frequency that the algorithm is run in each UAV-BS) could also be tuned in order to have a higher or lower resolution, at the trade-off of energy and complexity at each UAV-BS as well as based on the assumed user mobility model.

Algorithm 4: Proposed Solution

```
Initialize UAV-BS locations
Initialize Q-Tables
for Every episode do
    while Stopping criteria not met do
        UAV-BS selects action with largest Q value (ε-greedy)
        Allocate Users
        Observe reward, r_{t+1}
        Update next state
        Update Q-Tables, via (2.1)
    end
    if UAV is not in best position then
        Move to best observed reward
    end
    else
        UAV stay in the same position
    end
    Allocate users
    Record metrics
end
```

4.3.2 Metrics

In order to evaluate the proposed strategy, the metrics considered are the percentage of users in outage D_{u} and the average user throughput dissatisfaction D_{τ} per total number of users. The percentage of users in outage is given by

$$D_{\mathrm{u}} = 100 \cdot \frac{N_{\mathrm{o}}}{N_{\mathrm{u}}}, \tag{4.10}$$

where N_{o} is the total number of users in outage and is defined as

$$N_{\mathrm{o}} = N_{\mathrm{u}} - \sum_{j=1}^{N_{\mathrm{b}}} U_j. \tag{4.11}$$

By its turn, D_{τ} is computed considering all the users which have a throughput below his/her requirement, τ, such that

$$D_{\tau} = \frac{1}{N_u} \sum_{y \in \Psi} \frac{\tau_y - T_y}{\tau_y}, \tag{4.12}$$

where T_y and τ_y are the allocated and required throughput of user y, respectively, and Ψ is the set of users with $T_y < \tau_y$. In addition, if a user has not been allocated to any cell, it is considered dissatisfied and his/her perceived throughput

is assumed to be $T_y = 0$.

4.4 Results

4.4.1 Simulation Scenario

In order to showcase the effectiveness of the proposed solution, a simulation scenario in MATLAB has been built. It is considered that a network operator has a previously fully functional network in the area, composed of a macro and several small cells. However, a disaster happens and the previous network has been fully destroyed, with only parts of its original backhaul connection available, so the operator has to choose other means in order to restore connectivity as fast as possible.

For this scenario, it is considered that the operator chose to deploy a macro cell in a temporary fixed infrastructure, as for instance in a truck, in a position similar to the original one. Since it might be difficult for the operator to deploy the macro BS in its original position, either due to debris or blockages, in the simulations the truck containing the macro BS is positioned at its initial position added to an offset depending on a random distribution. In addition, drones are also deployed in the network to perform the role of the previous small cells. Regarding the drones positioning, several approaches are tested, such as fixing the drones in random positions, fixing the drones in a circular manner around the macro cell, fixing the drones in the previous location of the small cells, and, lastly, deploying movable and intelligent drones using the proposed Q-Learning solution. The proposed solution is compared to those baselines as there are no other methods, to the best of the author's knowledge, that perform the optimization of drone positioning in an emergency scenario, considering both RAN and backhaul constraints. The only other similar approach, the PSO based scheme in [207], would not be a feasible and appropriate solution to the proposed scenario, as it is not able to cope with the changes in the environment, nor scalable enough to be an adequate solution to a real situation. A summary of the simulation parameters is shown in Table 4.1 and Figures 4.4 and 4.5 show the simulation scenario setup in MATLAB.

The Environment

The simulated scenario consists of an urban area of 1km^2 ($L = 1$ km), following the model and parameters described in [223]. In this area, 768 users are scattered, some belonging to rescue teams and others consisting of regular users. Further-

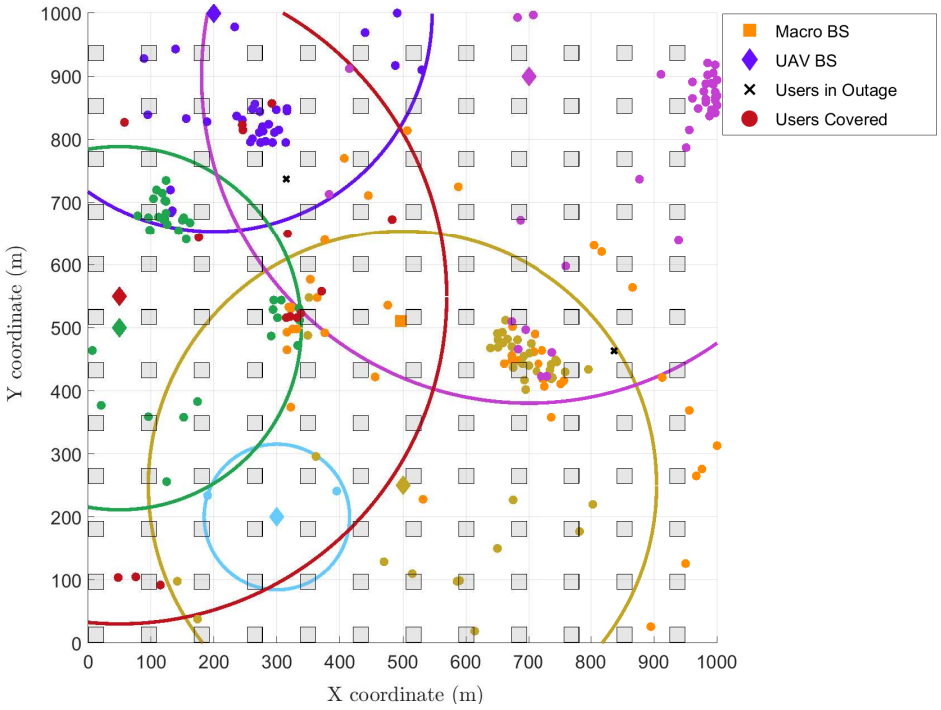


Figure 4.4: Top view of the simulation scenario. The macro cell, in orange, is positioned near the centre of the area, while the drones, are shown as coloured diamonds. The UAV-BSs coverage radius is represented as the coloured circles and users served by the BSs (either truck BS or UAV-BSs) are displayed with different colours.

more, different users have different characteristics in terms of both mobility and throughput requirements. As the throughput can be calculated in terms of the perceived SINR, as in (4.8), user requirements are modelled as SINR requirements instead of throughput requirements. Table 4.2 presents the different types of users, requirements and mobility levels.

User Distribution

A third of the users is randomly distributed in the entire 1km^2 area, whereas the other two thirds are randomly assigned to hot spots. In order to generate the hot spots, a defined number of hot spots are artificially created, and every hot spot has the same amount of users.

Frequency Bands and Antenna

Regarding frequency bands, it is assumed that both drones and macro cell would share the same frequency band, meaning that drones and macro cell would interfere with each other and a frequency reuse factor of 1 is considered. However, in order to mitigate the interference between drones, it is also considered that

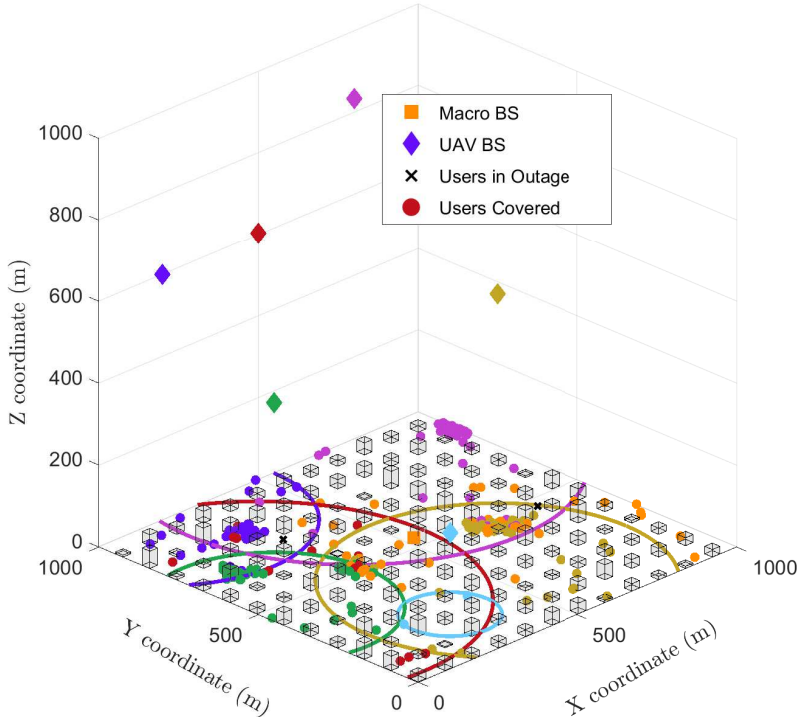


Figure 4.5: Isometric view of the simulation scenario. UAV-BSs adjust their 3D position in order to maximize the amount of users covered. As it can be seen, different UAV-BSs prefer different heights, in order to minimize interference between UAV-BSs while also maximizing their coverage.

each drone has a single antenna with an elevation and horizontal plane apertures of $\phi_{ap} = 60°$, which is a good approximation of commercially available antennas [226]. This means that each drone has a fixed radius of coverage, varying with its altitude, and that users out of that radius of coverage would perceive a very low signal coming from that drone.

RAN and Backhaul

In addition, the scenario also assumes that both macro and UAV-BSs are limited in both RAN and backhaul resources. Both macro cell and UAV-BSs are assumed to share a 10MHz bandwidth, which correspond to a capacity of 50 RBs, according to LTE parameters. Moreover, it is considered that the macro cell has an ideal backhaul [2,3] and that the UAV-BSs have a microwave link that connects to the macro cell.

Simulation

In order to implement the Q-Learning solution to the problem, first, a discretization of the proposed environment is performed. In terms of user mobility, it is

Table 4.1: Simulation Parameters

Parameters	**Value**
Ratio of build-up to total land area, α	0.3 [221]
Average number of buildings, β	500 buildings/km^2 [221]
Scale parameter for building heights, γ	15m [221]
ξ LOS	1 dB [223]
ξ NLOS	20 dB [223]
Side of the square area, L	1km
Drone X-axis step	50m
Drone Y-axis step	50m
Drone Z-axis step	100m
Minimum Drone height	200m
Maximum Drone height	1,000m
Low mobility Users X-axis step	3m
Low mobility Users Y-axis step	3m
Low mobility Users Z-axis step	0m
High mobility Users X-axis step	10m
High mobility Users Y-axis step	10m
High mobility Users Z-axis step	0m
Number of users, N_{u}	768 [2,3]
User height, h_{u}	1.5m
Ratio of rescue team users	20%
Number of hot spots	16
Number of UAV-BSs	16
Ratio of users in near hot spots	2/3 [2,3]
Macro BS EIRP	0 dBW [2,3]
Macro BS height, h_{B}	20m
UAV-BS EIRP	-3 dBW [225]
UAV-BS antenna directivity angle, ϕ_{ap}	60° [226]
RBs in Macro Cell	50 [2,3]
RBs in UAV-BSs	50 [2,3]
Macro cell backhaul capacity	100Gbps [2,3]
Microwave backhaul capacity per drone	37.5Mbps/drone [2,3]
Bandwidth of one RB	180kHz [2,3]
Carrier frequency, f_{c}	1GHz
High SINR requirement	5dB
Low SINR requirement	0dB
Total number of episodes	100
Number of independent runs	100
Max iterations per episode, $\mathrm{Max}_{\mathrm{it}}$	1,000
Max iterations, same reward, $\mathrm{Max}_{\mathrm{it,r}}$	100
Min iterations per episode, $\mathrm{Min}_{\mathrm{it,r}}$	200
Learning Rate (λ)	0.9
Discount Factor (ϕ)	0.9

Table 4.2: User characteristics

	User Types	
	Rescue team	**Regular**
Mobility (metres/iteration)	10	3
SINR (dB)	0 or 5	0

considered that users could move in steps of 1 meter in any direction. It is also considered that the user mobile phone distance to the ground is of 1.5 meter height. Regarding the UAV-BSs movements, the aerial space is discretized in steps of 50 meters in the horizontal plane (X and Y dimensions) and in steps of 100 meters in the vertical domain (Z dimension). Drones could then either move in this space in all three dimensions or stay still.

The simulation is ran for 100 independent runs, each with a total of 100 episodes (snapshots of the network). First, however, before the drones start moving and determining their best positions, an initialization process is performed. Before the simulation begins, user positions, requirements and mobility levels are generated. Then, a certain number of hot spots are generated with an equal amount of users per hot spot and a fixed number of drones is also positioned in the system, according to the scenario being evaluated. In addition, the backhaul of the drones is also initialized according to the microwave backhaul parameters and each drone is assumed to have a random initial backhaul load, of up to 10% of its initial capacity, due to control channels.

The proposed solution is compared to 3 different positioning strategies, mainly:

1. Deploying the drones in fixed random positions;
2. Deploying the UAVs in a fixed circular formation centred around the macro BS at evenly spread angles;
3. Deploying the drones in the locations of the previous hot-spots of the destroyed network.

For the fixed position scenarios, the drones remain in their locations for all episodes, while users move around according to their mobility behaviours. On the other hand, for the Q-Learning approach, the UAV-BSs try to find the best possible position for every episode. For this to be possible, however, each episode is also divided into iterations. For every iteration the drones move around the environment looking for the best positions in the system, and during this process the users are considered to be static. Every drone performs a certain amount of iterations, according to the stopping criteria of the algorithm and determines

the best position for that episode. After all drones stop moving, the episode is finished, all metrics are recorded and user positions are updated, so that the mobility of users can be taken into account between episodes. Lastly, whenever a new episode begins, the UAV-BSs start from the previous learned position and with the previous computed Q-Tables, in order to explore the correlation between different snapshots of the network. This process is then repeated and the results are averaged out between different runs of the algorithm.

4.4.2 Numerical Results

Figure 4.6 shows the average number of users in outage per episode for each of the considered strategies. As it can be observed, the Q-Learning approach yields the best results, resulting in around 2% of users in outage after 100 episodes and down to less than 5% after only 10 episodes. In addition, it can also be seen that both the random fixed positioning strategy as well as the circular positioning strategy yield very poor performance in terms of covered users, having more than 50% of users in outage at any given episode. Additionally, as the metrics are computed at the end of the episodes (after the UAV-BSs have performed their movements) it is natural that the Q-Learning and random positions curves start at different values, as in the former case, the UAV-BSs move, while in the latter they do not. Keeping the drones fixed at the location of the hot spots also results in a poor performance, albeit better than the random and circular strategies. This result emphasizes the importance of having a movable solution, because as users move through the network, the proposed solution is able to detect and track user movement, learning the best positions to be in for every episode. In addition, the proposed Q-Learning solution also shows that it is able to provide coverage and service whenever and wherever it is needed.

Figures 4.7 and 4.8 show the RAN load of the drones (averaged over all drones) and the macro cell, respectively, per episode. Regarding the Q-Learning strategy, the RAN load of the drones increases from below 80% up to almost 90%, as can be seen in Figure 4.7. This indicates that the stopping criteria being reached is not the one related to having full capacity, but rather the one in which the reward does not improve after a certain number of iterations. Moreover, this happens mainly because there are more RAN resources than the minimum required to serve all users in the network, such that the reward does not vary for longer than 100 iterations. In addition, looking at Figure 4.8, it is possible to observe that the load on the macro cell almost does not vary as the UAV-BSs find better positions in terms of reward (served users overall). Although not being completely loaded as the other approaches, Figure 4.8 shows that the UAVs try not to serve users

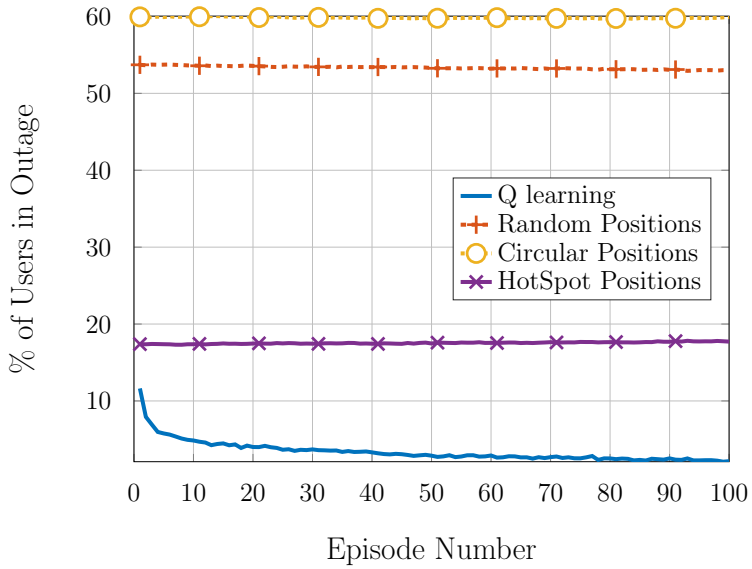


Figure 4.6: Average number of users in outage per episode.

already served by the macro cell, despite the UAVs "stealing" a couple of users. This indicates that the drones are finding, in their majority, users in outage that could not be served by the macro cell.

This behaviour of following and allocating users that would be in outage can also be deducted from Figures 4.6 and 4.7, by comparing the trend of the curves for users in outage and drone load. Note that both curves have inverted trends, indicating that the UAV-BSs are using their spare capacity to provide coverage to a wider number of users. On the other hand, Figures 4.7 and 4.8 also show that the other positioning strategies have almost no variation in the curves of drone RAN load, since they do not adapt to the mobility patterns of users. Furthermore, because the RAN load on the UAV-BSs is relatively low, the macro cell is always operating at full RAN capacity in these cases.

Figures 4.9 and 4.10 show the average dissatisfaction per user in terms of throughput for users with low and high requirement, respectively. As it can be seen, the Q-Learning solution, despite not being explicitly programmed to mitigate user dissatisfaction, is the best performing strategy by a large margin. This is mainly due to the fact that users out of coverage are considered 100% dissatisfied. The performance regarding throughput satisfaction could even be improved if the reward considered this metric, however since this is not the main goal of the proposed solution (nor the main requirement for the type of application), the Q-Learning approach does not optimize user satisfaction.

Figure 4.11 shows the average backhaul throughput for the UAV-BSs. As it can be seen, there is not a large variation of the parameters in different episodes but the important thing to note is that the backhaul capacity of the UAV-BSs

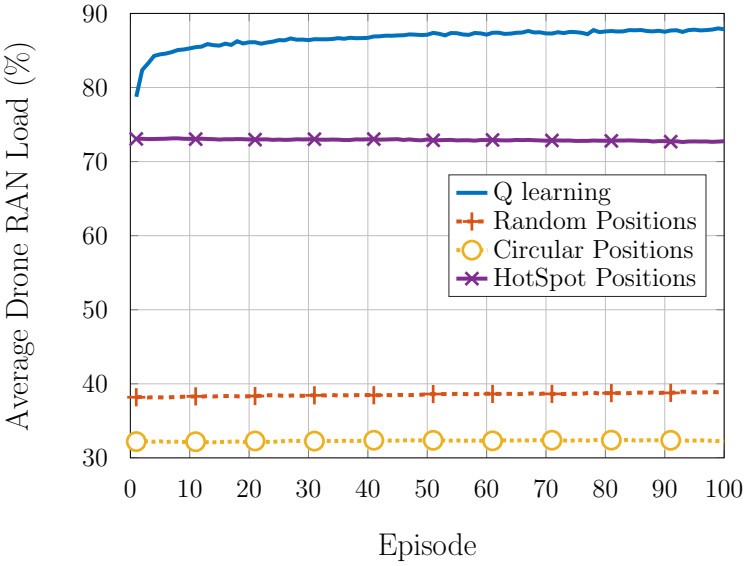


Figure 4.7: Average UAV-BS RAN load per episode.

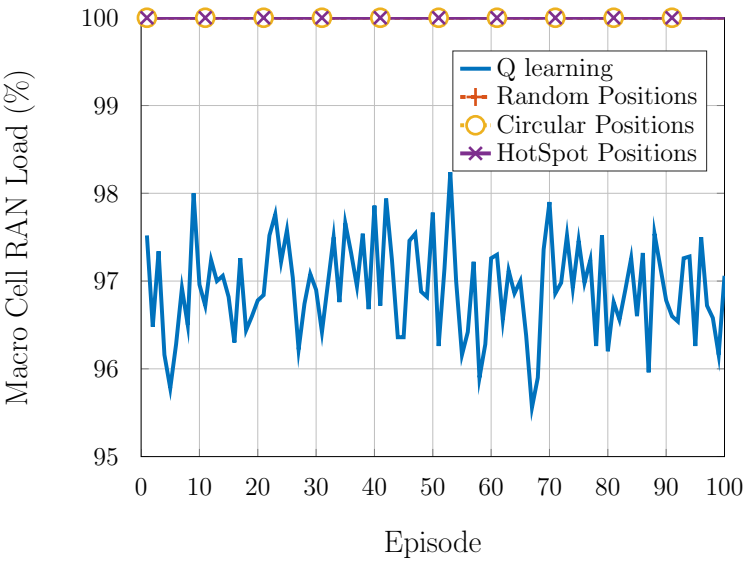


Figure 4.8: Average macro cell RAN load per episode.

is not being exceeded, showing that the throughput bottleneck is the SINR. In addition, when comparing to the maximum possible backhaul capacity of each drone, it can be seen that the Q-Learning strategy is the one that best utilizes the backhaul resources of the system, while the other strategies do not use the backhaul resources very well, leaving more capacity unused.

Lastly, the impact of different learning rates, λ, is also investigated and their influence is illustrated in Figure 4.12. As expected, the learning rate exerts some influence in the convergence rate of the algorithm, as higher λ leads to better results, as can be observed. For instance, considering $\lambda = 0.1$ results in 4% of the users being in outage after 100 episodes, whereas increasing the value of λ

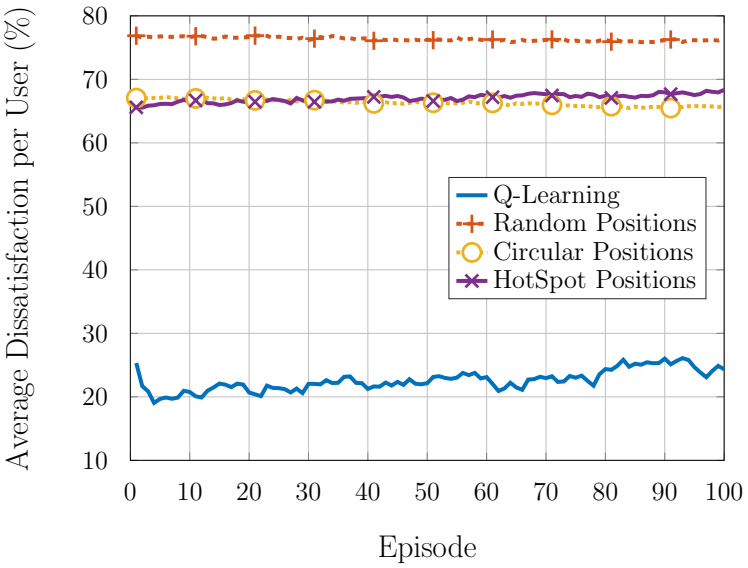


Figure 4.9: Average dissatisfaction of users with low throughput requirement.

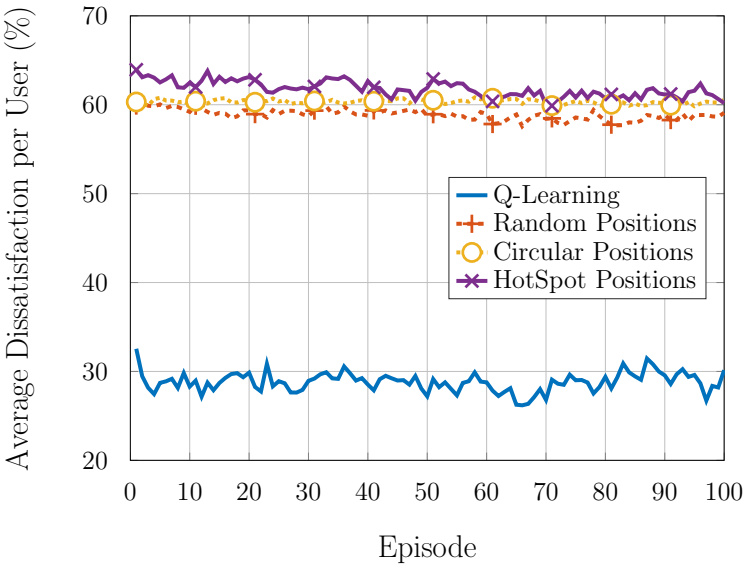


Figure 4.10: Average dissatisfaction of users with high throughput requirement.

gradually improves performance. This difference can be explained due to the fact that the drones learn less from the environment when smaller λ are considered, and thus are less able to adapt to the changes in the environment, such as user mobility. Hence, a value of $\lambda = 0.9$ is chosen.

4.5 Summary

In order to provide service whenever large-scale natural disasters happen, it is crucial that network operators have adaptable and intelligent solutions at hand.

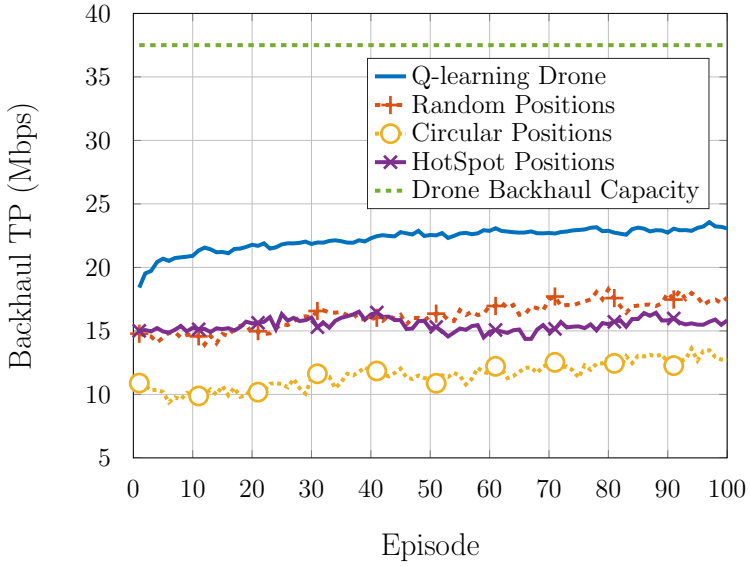


Figure 4.11: Average backhaul throughput for the drones per episode.

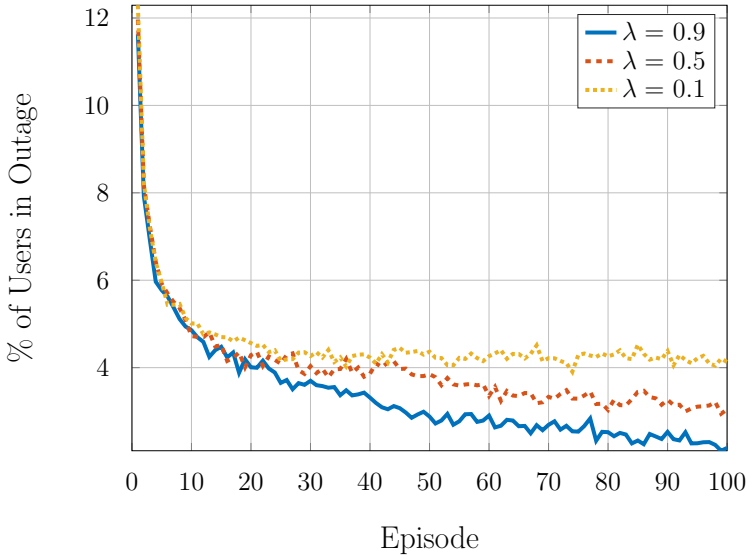


Figure 4.12: Users in outage per episode considering different learning rates for the Q-learning positioning strategy.

With that in mind, new solutions have to be created, as conventional approaches and regular BSs might not be suitable or fast enough in order to provide service in such emergent situations. Hence, one possible enabler for ECNs is the deployment of intelligent drone BSs, as they can provide coverage whenever and wherever needed, due to their mobile characteristics.

Based on the advantages that UAV BSs can provide for future emergency mobile networks, in this chapter a RL based approach was envisioned and developed to jointly optimise the position of multiple UAV BSs limited by network constraints. The proposed algorithm works in real-time in order to determine

the 3D positions of each UAV and it is based on a distributed implementation of Q-Learning, with each UAV having its own Q-Table. Results show that the intelligent based solution outperforms other methods in terms of coverage (minimizing the number of users in outage) and user satisfaction, while also utilizing better network backhaul resources. These results showcase the importance that mobile BSs can provide in future emergency networks, as they are capable of learning 3D positions without any knowledge of the underlying environment, only based on network constraints.

Chapter 5

Intelligent UAVs for Pop-up Networks

5.1 Introduction

The fifth generation, 5G, and beyond are expected to overcome limitations of current networks, and also push their performance to a next level enabling new applications that are unimaginable today [8]. For this to be possible, however, 5G is expected to see various improvements in all network layers. One such improvement that has seen increased attention in recent years is the utilization of UAVs as aerial BSs. As aforementioned in Chapter 4, due to their mobility, flexibility and LoS communication characteristics, UAVs are envisioned to play a key role in future communications, making mobile networks much more flexible, adaptable and agile [209, 211, 228, 229]. However, as demonstrated in [228, 230, 231], despite recent extensive research, several issues still remain, such as:

- Finding the optimal deployment of multiple UAV BSs;
- How to handle the interference created by UAVs;
- Seamless (dis)integration of aerial BSs to core network;
- Resource allocation and management in heterogeneous air-ground networks;
- UAV backhaul and HO management.

As such, in this chapter a solution to the problem of determining the optimal 3D deployment of multiple UAVs constrained by network conditions is proposed. In this context, a pop-up network scenario is investigated. Pop-up networks consist of networks that can form spontaneously, or randomly and can consist of several users concentrated in a relatively small areas. These networks are

quite common when certain events take place, such as musical concerts, sporting events, or during national holidays. Although sometimes predictable, such as in the cases of scheduled events, in other occasions these networks might "pop-up" in a completely random manner, hence the name. Nevertheless, pop-up networks are usually a problem for mobile operators, since deploying fixed infrastructure just for the duration of the events might not be cost-effective nor feasible [8]. This, by its turn, causes several problems, specially in terms of congestion and user satisfaction, as the ground network is not capable of handling the excessive capacity required.

Thus, in this chapter a solution to the problem of coverage and capacity provisioning in the case of pop-up networks is investigated. This is done by utilizing UAVs coupled with RL algorithms, in order to provide the necessary service whenever and wherever needed. Differently than the previous chapter in which a ground network is assumed to be completely destroyed, with only a backhaul connection remaining, in this chapter a more complex scenario is investigated, in which a ground network is still present, composed of a macro cell and several hot-spots, which, by its turn cause additional interference in the flying network and vice-versa. In addition, both aerial and ground networks have limiting constraints, such as in terms of RAN and backhaul and users present different requirements as well as different levels of mobility. Thus, due to the relative simplicity of the scenario considered in Chapter 4, when compared to the scenario investigated in this Chapter, the utilisation of simpler RL algorithms to position multiple UAVs were enough. In addition, VFA could also have been studied in Chapter 4, however despite VFA being really powerful, it also adds more complexity to the system, in terms of the function that needs to be learnt as well as in terms of convergence properties [10]. As such, based on the complexity of the proposed scenario evaluated in this chapter, the utilisation of RL combined with VFA is proposed. The proposed solution, utilises SARSA with VFA, and its results are compared to the baseline of the traditional Q-Learning. Results show that despite Q-Learning performing well, its limitations in terms of memory and generalization capability hinder its performance. Thus, when compared to SARSA with VFA, it can be clearly seen that VFA is more robust and can outperform traditional RL in all metrics considered.

5.1.1 Literature Review

Although some issues related to the deployment of UAVs in mobile networks have been addressed in recent publications, [39, 205, 207–209, 211–216, 223, 229–246], most of the solutions involving UAVs as aerial BSs are scenario specific, consisting

of analytical or heuristic approaches to a particular problem. This is a huge issue, as future networks are expected to be flexible, autonomous and adaptive [8].

For example, the authors in [230] develop an aerial BS deployment framework to overcome network overload based on three different scenarios alongside a prediction scheme to solve the positioning problem, by gathering data from the network and users. In [233], the authors attempt to position multiple UAVs based on network demand and develop two different cost functions to match UAVs to a specific area based on user demands and positions. On the other hand, authors in [211, 231], present reviews on challenges and opportunities in UAV-based networks, while [231] also investigates the performance of a multi-tier UAV network in terms of spectral efficiency and connected users. In [234], two mathematical models are developed to determine the performance of UAV networks in terms of data rate and resource allocation when flight-time constraints are considered. In [235], a novel 3D mobile network architecture and mathematical models for UAV deployment and cell association considering minimal latency are developed. Mozaffari *et al.* in [236] also develops a mathematical model that utilizes UAVs as an aerial antenna array to minimize user service time and improve spectral efficiency. In [237] the deployment of aerial BSs in the presence of a ground network is investigated and a model is developed to determine how many UAVs are necessary and where to position them. In [239] a mathematical model to determine the best 3D position of a single UAV is developed with the objective of maximizing network revenue, while in [240] the authors mathematically determine the best 3D position of a UAV by an exhaustive search using a novel low-complexity algorithm.

In [241] UAVs are used as relays to assist in emergency scenarios. In [242], a mathematical model to find the optimal position of a UAV, which leads to the lowest power consumption is proposed. Al-Hourani *et al.*, in [212], present an analytical approach to determine the optimal altitude of a UAV, providing maximum coverage ground radius and in [213] the authors develop a mathematical framework to optimize downlink coverage and transmit power considering the altitude of a UAV. In [209] an analytical framework when UAVs are deployed in a device-to-device network is developed. The coverage and throughput impacts are analysed and tests in static and mobile UAV scenarios are considered. Ahmadi et al., in [208], propose a novel network architecture with multiple UAVs in different altitudes and mathematically formulate the optimal placement of multiple UAVs, whereas in [216] the authors optimize UAV placement considering backhaul constraints. Moreover, in [243] the authors also evaluate the performance of UAV backhaul, by considering a novel Free-Space Optics backhaul link for aerial

BSs. Furthermore, in [214], an optimal placement algorithm for aerial BSs is proposed, such that the number of users covered is maximized, while using minimum transmit power.

On the other hand, other solutions such as [39, 207, 229, 232, 238, 244–246] utilize machine learning in order to optimize aerial network parameters. For example, in [229], a Gaussian mixture model is proposed to determine the optimal placement of UAVs considering minimum UAV power consumption. In [39] a RL solution to position multiple UAVs in an emergency communication scenario, in which the previous network infrastructure is destroyed, is investigated with the primary objective of allocating as many users as possible. Chen et al., in [232], propose an echo state network to position UAVs and determine which contents to cache at each aerial BS. In [207], a particle swarm optimization is used to find the 3D placement of multiple UAVs so that all users are served. Ghanavi *et al.*, in [244], propose a RL solution to place a single UAV in order to enhance users' QoS. Lastly, [245] proposes a neural network to provide reliable connectivity and secure UAV links, whereas [246] proposes an echo state network for path planning of a single UAV along a fixed network.

Issues with Current Solutions

Although analytical solutions are important, in order to provide the necessary flexibility and adaptability for current and future mobile networks and to fully enable the concept of SON, analytical and scenario specific methods are not enough, and, more flexible and intelligent solutions are needed [8]. This occurs because several analytic methods rely on strong assumptions, such as the knowledge about how many users are connected to the network at a given point and their positions. In addition, the 3D placement problem of multiple aerial BSs is a NP-hard problem, due to the complexity of channel model, interference between multiple UAVs and ground BSs, and movement of users [237, 239]. Furthermore, if exhaustive search methods are utilized, the complexity is $O(2^n)$ [243], in which n is the number of users in the network, which is also not feasible for typical networks with hundreds of users.

On the other hand, intelligent methods found in the literature also have gaps. For example, [229] utilizes a Gaussian mixture model based on previous network information (number and position of users at previous time-steps and network traffic). Chen et al., in [232], consider a neural network that needs to be fed with certain inputs, which rely on having a lot of information about the network. Similarly, Challita et al., in [245], also require user context information in order to train a neural network based UAV solution. The proposed approach in [207],

due to its inherit heuristic nature, is also not suitable and might not be able to cope with a dynamic changing environment. Another work, [246], utilizes deep RL to perform path planning in a fixed ground network, in which the UAV must find the best path to traverse and reach a destination. However, for this case a simple model of a fixed ground network, with fixed users was considered and the objective was only to minimize the interference along the path.

Motivation

Based on the aforementioned issues, it is clear that analytical solutions are not suitable for the on-line deployment and positioning of UAVs. This occurs because analytical solutions require extensive knowledge about the environment and the network, such as the number of users connected to the network, user-cell association, user requirements and positions. As such, whenever one of these parameters changes, analytical solutions need to evaluate the whole system again in order to determine the new optimal solutions.

Furthermore, not only analytical solutions suffer from their drawbacks, most of the intelligent state-of-the-art solutions also have their own limitations, as they rely on training data or information that might not be easily available [232, 245, 246]. Moreover, recent regulations regarding privacy may hinder the use of specific user information without consent, even when it is available. This fact highlights again that solutions that do not depend on a specific model or that do not require the collection of training data beforehand are preferred. In this sense, model-free solutions, i.e. approaches that interact with the network and determine by themselves the optimal or best solution by experience, such as those based on RL are needed.

For example, in our previous work, in [39], a model-free RL approach was utilized in order to determine the best position of multiple aerial BSs and achieve optimal UAV 3D placement. It was shown that UAVs were capable of quickly adapting and finding optimal positions to deliver almost 100% coverage without any knowledge about the underlying network (only based on the reward of how many users were covered in total). However, [39] also has its limitations, as no interference between UAVs and ground BSs were considered and the mobility of users was assumed to be random. In addition, in real situations, conventional RL algorithms, as in [39], might not be suitable for complex problems, due to their memory limitations and lack of generalization abilities. As such, more efficient RL approaches are also necessary [10].

5.1.2 Objectives and Contributions

The utilization of UAVs as mobile BSs coupled with RL algorithms in order to determine their optimal positioning is proposed. The problem of coverage and capacity enhancement in a scenario of pop-up networks, such as when events happen and the ground network cannot cope with the increased demand, is investigated. A network model consisting of both terrestrial and aerial BSs is considered and the goal is to provide coverage to as many users as possible. Two model-free RL techniques are explored: Q-Learning and SARSA with VFA. In this scenario it is shown that the two intelligent solutions for the deployment of UAVs are highly beneficial, as they can learn by themselves the best positions without any information or previous knowledge from the network, such as how many users there are at a given time, their positions or their association, highlighting the applicability of RL solutions in real situations. Results show, that both RL solutions can minimize network outage and user dissatisfaction levels. Lastly, results also show that more robust RL algorithms, those based on VFA, are able to outperform conventional RL solutions, while also being more flexible and requiring less memory.

In addition, a small simulation scenario is also developed in order to demonstrate that RL solutions are able to perform as well as analytical approaches. In this small scale scenario, both proposed RL algorithms are evaluated and compared against an exhaustive search approach, and the results clearly show that RL based solutions are able to provide significant gains with respect to complexity and computational time, while achieving near-optimum solutions. This small scenario also shows the clear advantage of VFA when compared to conventional RL, independently of the approximation function being utilized, as even a linear function is able to outperform conventional RL strategies. Lastly, this small scenario also enables the analysis of the proposed RL algorithms complexity and convergence properties. The contributions of this scenario can be summarized as:

- Implementation of two RL algorithms in a pop-up network scenario, showing that they are suitable for UAV positioning, while providing coverage and minimizing user dissatisfaction;
- Analysis of the convergence and complexity of conventional and VFA based RL algorithms;
- The deficiencies and limitations present in conventional RL algorithms, such as their constraints in terms of memory and generalization ability are discussed and results show that VFA is able to achieve more robust and better solutions.

5.2 System Model

Based on the aforementioned applications envisioned for UAVs, the problem of coverage and capacity enhancement in a pop-up network scenario is investigated. Whenever unexpected events happen, such as fairs or open markets, there is usually a large concentration of people in a relatively small area, resulting in many users being out of coverage as the network becomes congested. Given that these events are temporary, operators often consider long-term or fixed solutions infeasible or cost-ineffective, leading many users to experience a poor QoS [228, 230]. Hence, the deployment of intelligent aerial BSs is envisioned to provide the additional network coverage necessary when unexpected big events or crowded situations happen. By coupling UAVs with intelligent algorithms, autonomous positioning can be achieved requiring minimal human intervention, while also attaining the desired flexibility and agility needed for self-organizing network solutions [8].

5.2.1 Baseline Network Model

Similar to the previous scenario, a Manhattan grid of a city was considered (an L by L square area representing a city with buildings uniformly distributed), based on the ITU-R parameters and following the model presented in [223]. In this squared area, a heterogeneous network scenario is considered, in which a set $\mathbb{M}$ containing m three-sectored macro BSs, $\mathbb{M} = \{M_1, M_2, ..., M_m\}$, is positioned and a set $\mathbb{K}$ containing k small cells, $\mathbb{K} = \{K_1, K_2, ..., K_k\}$, is randomly and uniformly distributed in the area. In addition, u users are randomly and uniformly distributed, with 2/3 being randomly and uniformly positioned in the vicinity of of the k small cells and the other 1/3 randomly and uniformly positioned across the entire squared area [2].

In addition to the ground network, a set $\mathbb{V}$ containing v UAVs is deployed in order to enhance the capacity of the mobile network, $\mathbb{V} = \{V_1, V_2, ..., V_p\}$. Similarly to the previous scenario, UAVs are assumed to have a directional antenna pointing downwards with an aperture angle of ϕ_{ap}, which is responsible for providing connectivity to ground users. Lastly, a microwave backhaul, which connects the UAVs to the ground macro BS to provide network service to the aerial platforms, is also assumed.

5.2.2 Pop-Up Network Model

Considering the pop-up network scenario, it is assumed that an event happens in a random part of a city at a given time, and that a random percentage of users, between 60% to 80%, attends the event. Whenever the event starts, it is assumed that this random percentage of users moves towards a randomly assigned position in the event, whereas the remaining users are assumed to have random mobility. The event is assumed to be in a circular area of radius R_{e} and once the users get to the event it is assumed that they move freely and randomly inside the event area. In terms of mobility, it is also considered that users have different mobility levels, with some users moving faster and others slower. Moreover, users can also have two different throughput requirements, being able to request either high or low data rates from the network.

5.2.3 Channel Models

The path loss of the macro BSs ($H_{u,m}$) is defined according to the Okumura-Hata model (as defined in (4.3)), and the path loss of small cells is given by [4]

$$H_{u,k} = 140.7 + 36.7 \cdot \log_{10} d_{u,k}. \tag{5.1}$$

where $d_{u,k}$ is the distance between a user and a small cell.

Lastly, the path loss of the aerial BSs is given according to [223], following the same model as in (4.6).

5.2.4 User-Cell Association

For a user to be associated to a BS and allocated resources, the same formulation as in Chapter 4 is followed, in which the RSRP, in dB, of user u and any BS b, with $b \in \mathbb{B} = \{\mathbb{M} \cup \mathbb{K} \cup \mathbb{V}\}$, is given by

$$R_{u,b} = \mathrm{EIRP}_b - H_{u,b} \quad \forall b \in \mathbb{B}, \tag{5.2}$$

where EIRP_b is the equivalent isotropically radiated power and $H_{u,b}$ is the path loss, both with respect to BS b.

The SINR of user u at BS b (in dB) is calculated as

$$\mathrm{SINR}_{u,b} = R_{u,b} - \left(N + \sum_{i=1, i \neq b}^{n} R_{u,i} \right). \tag{5.3}$$

If the user has successfully connected to a BS (its SINR was above a certain

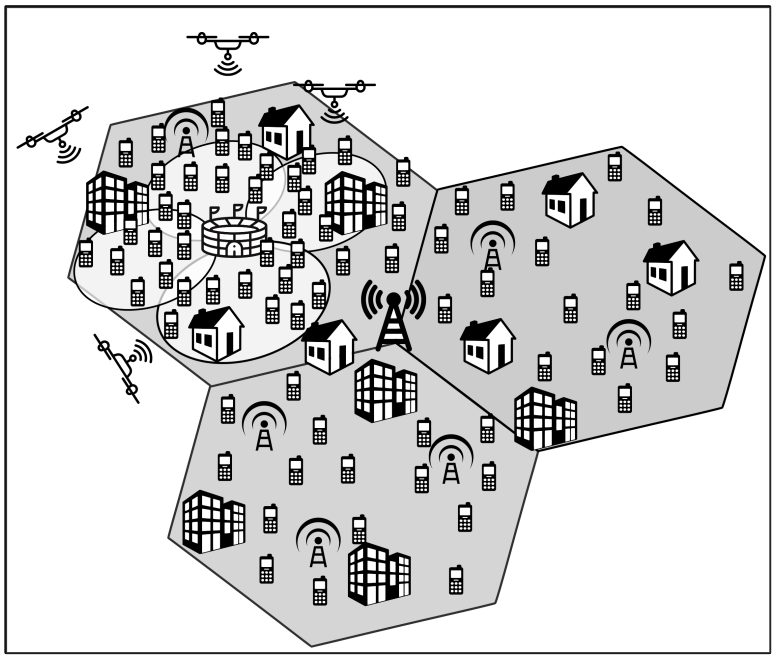


Figure 5.1: Model of the pop-up network scenario considered, with $m = 1$ macro BS, $k = 6$ small cells, $v = 4$ UAVs and $u = 70$ users.

threshold and a RB was available) its throughput, $T_{u,b}$, is estimated according to Shannon's capacity, as in (3.5) and the backhaul throughput demanded by users allocated to a specific BS b is given by

$$\tau_{u,b} = \rho \cdot T_{u,b}, \tag{5.4}$$

where ρ represents a backhaul overhead factor [2].

Figure 5.1 shows a representation of the model of the heterogeneous network considered in this work, including the three sectored macro BS, the small BSs and the aerial BSs, as well as a stadium that hosts temporary events from time to time.

5.3 Problem Formulation

The application of two different RL algorithms is envisioned in order to position multiple UAVs in a pop-up network scenario. The objective of the proposed schemes is to maximize the total number of users covered by both the ground BSs as well as aerial BSs. This could be done in a centralized manner, however this would require an additional layer of control and synchronization, which could not be feasible due to the increase in latency, control and network overhead. As such, a distributed approach is preferred, in which each UAV, v, is an agent of the

RL framework, and the network is the environment. In the case of a distributed solution, the global optimization problem is divided into smaller sub-problems of trying to allocate the maximum number of users per UAV. Since UAVs interfere with each other, as well as the ground network, this approach is valid, as if UAVs position themselves too close to each other or to other ground BSs, interference will increase, minimizing the number of users allocated to the network. Furthermore, if UAVs also position themselves too close to the ground BSs, the UAVs will "steal" users from the ground network, leading to poor resource allocation (as most ground BSs would end up with their RAN underutilized). As such, a strategy to maximize the total amount of users connected to both networks (ground and aerial) is adopted.

A distributed approach to the problem of positioning multiple UAVs in a scenario of pop-up networks is formulated. The goal of the system is to maximize the number users allocated to the network (either ground or aerial). Thus, the problem can be formulated as:

$$\begin{aligned}
\underset{u_b}{\text{maximize}} \quad & \sum_{i=1}^{n} u_b && \text{(5.5a)}\\
\text{subject to} \quad & x_{v,\min} \leq x_v \leq x_{v,\max},\ \forall\ v \in \mathbb{V}, && \text{(5.5b)}\\
& y_{v,\min} \leq y_v \leq y_{v,\max},\ \forall\ v \in \mathbb{V}, && \text{(5.5c)}\\
& z_{v,\min} \leq z_v \leq z_{v,\max},\ \forall\ v \in \mathbb{V}, && \text{(5.5d)}\\
& \text{EIRP}_b \leq \text{EIRP}_{b,\max},\ \forall\ b \in \mathbb{B}, && \text{(5.5e)}\\
& \text{RB}_b \leq \text{RB}_{b,\max},\ \forall\ b \in \mathbb{B}, && \text{(5.5f)}\\
& \text{SINR}_{u,b} \geq \text{SINR}_{u,b,\text{req}},\ \forall\ u, b \in \mathbb{U}, \mathbb{B}, && \text{(5.5g)}\\
& \sum_{i=1}^{n} \tau_{i,b} \leq C_{b,\max},\ \forall\ b \in \mathbb{B}. && \text{(5.5h)}
\end{aligned}$$

where x_v, y_v, and z_v are the X, Y, Z coordinates of UAV v, respectively, R_b is the number of RBs of BS b, and $R_{b,\max}$ is the maximum number of RBs a BS can have.

As it can be seen, maximizing the total number of users allocated in the network, in (5.5), is subject to 7 different constraints. The first three, (5.5b), (5.5c), (5.5d) are responsible for restricting the UAVs movements in the square area and at certain heights. Due to simulation constraints, these values are assumed to be discrete and have fixed steps on each axis $x_{v,\text{step}}, y_{v,\text{step}}, z_{v,\text{step}}$. The fourth constraint, (5.5e), states that both aerial and ground BSs should not have a transmit power more than their maximum power, given by $\text{EIRP}_{b,\max}$. Constraint, (5.5f), concerns the RAN of the BSs. It states that any BS b should not allocate

more users than they are allowed to, given by $\mathrm{RB}_{b,\max}$. On the other hand, (5.5g) guarantees that users will only be associated to a BS if their perceived SINR, $\mathrm{SINR}_{u,b}$ is greater or equal than the required SINR, $\mathrm{SINR}_{u,b,\mathrm{req}}$. In other words, users have certain requirements in terms of throughput and the BSs need to attend those requirements. Lastly, (5.5h) guarantees that the backhaul throughput of any BS b does not exceed the maximum throughput allowed, $C_{b,\max}$.

5.4 Proposed Solution

In order to solve (5.5), two model-free approaches based on RL are proposed to find the 3D positions of multiple UAVs in the context of pop-up networks. The first approach considers the Q-Learning algorithm. However, as it as previously highlighted, conventional RL have problems in terms of memory and generalization capabilities, as such, in order to overcome this issues, SARSA with linear VFA is also proposed.

5.4.1 Reinforcement Learning (RL) Model

In this work, a distributed RL algorithm is considered, where v independent agents are deployed. Below a description of each component of the RL model is presented.

Environment

The heterogeneous network, its users and the buildings (which directly impact the path loss between users and UAVs). The environment is responsible for generating information which will be utilized by the agent, such as: the movement of users; user-cell association; LoS and NLoS conditions; RSRP and SINR of users; interference between ground-ground BSs and ground-aerial BSs;

Agents

The UAVs, which leads to a multiple agent RL system.

States

Each agent makes its decision based on a signal coming from the environment, called the state, s_t. In this context, the states of each UAV are their 3D positions in space.

Actions

Each UAV can choose one action, a_t, out of seven possibilities. The action takes UAVs from their current state to the next and consist of moving in any direction (right, left, up, down, forward, backward) or do not move at all.

Policy

The UAVs utilize an ϵ-greedy policy with a decaying ϵ. This policy states that with a probability $p = (1 - \epsilon)$ the action that yields the maximum value known by the agent is chosen, whereas with probability $p = \epsilon$ a random action is chosen. This allows the UAVs to explore in earlier stages, and later on to exploit the information collected from the environment [10].

Reward

Since the main objective is to address the lack of coverage in a pop-up network scenario, the reward is defined as the total number of users connected to the network. Because the reward is based on user-cell associations, it ensures that each user perceives a minimum SINR, guaranteeing the desired QoS. Mathematically, the reward is expressed as

$$r_{t+1} = \sum_{i=1}^{n} u_b. \tag{5.6}$$

5.4.2 RL Methods

Q-Learning

The Q-Learning solution is similar to the one in Chapter 4 and the agents learn to move around in the environment using the update in (2.1).

SARSA with Linear VFA

Mathematically, SARSA's update is expressed as in (2.2). However, instead of utilizing a memory fixed action-value table, instead SARSA combined with VFA is proposed, so that generalization and more robustness can be achieved [10]. In terms of VFA, a linear function was chosen, such that

$$Q(s_t, a_t, \vec{\theta}) = \mathbf{F} \cdot \vec{\theta}, \tag{5.7}$$

where $\mathbf{F}$ are features of the model, represented by a row vector of dimensions $[1 \times n_f]$, n_f is the number of features, and $\vec{\theta}$ are the weights learned by the

algorithm to represent the Q-Function, given by a column vector of dimensions $[n_f \times 1]$.

In addition, for the algorithm to be capable of learning the best weights to represent its function, a GD approach was considered, in which the error is adjusted by a small amount in the direction of the gradient. The error in this case is given as in (2.3) and the GD update can be computed as in (2.6). However, since the action-value function is represented by a linear function, its gradient is equal to the features themselves, and thus (2.6) becomes

$$\vec{\theta} = \vec{\theta} + \lambda_{\text{GD}} \cdot e_t \cdot \mathbf{F}. \tag{5.8}$$

Furthermore, in terms of features, properties or characteristics of the network can be selected, such as the average RSRP or SINR of connected users, the total number of users connected to the network, the distance from one UAV to the others, etc. However, designing features is a difficult task and often varies for different problems, thus it is a good approach to experiment and try different features depending on the task being performed [10]. By including features in the model, the new improved RL algorithm can learn intrinsic patterns in the environment and translate them to values of the action-value function, learning which states are better to be in and which actions are better to take. This will translate in the algorithm learning better policies, as well as a better estimate for the action-value function [10].

5.4.3 Proposed Framework

Based on the aforementioned scenario, an intelligent distributed framework to position multiple UAVs in the case of a pop-up network setting was designed. Each UAV is considered to be an independent agent of the system and it is equipped with an intelligent algorithm (either Q-Learning or SARSA).

Due to the nature of the simulated environment, a discretization process is performed, in which UAVs choose between a finite number of 3D positions so that a limited number of states are evaluated and stored in the table, for the case of Q-Learning. In addition, simulation steps are considered, and are divided into three categories: runs, episodes, and iterations.

1. Runs correspond to different realizations of the environment, in which environment parameters are generated, such as initial user positions, mobility pattern, user requirements, small cell locations, which users and how many are attending the event. In every run all metrics and learning parameters are reset, so that the average performance of the system can be measured

and computed at the end, or in other words, runs are completely independent realizations of the environment;

2. Episodes correspond to different snapshots of the network, in which v UAVs move for a certain number of iterations, according to a stopping criteria. Since snapshots of the network are correlated, the parameters learned by the UAVs, such as the action-value functions and the weights θ, are kept between consecutive episodes, so that UAVs can exploit what they have learned in the past in future episodes;

3. Iterations correspond to the movements or actions of each UAV. It is assumed that UAVs move from the set of possible positions in every iteration. Lastly, each UAV can perform a different number of iterations based on its actions and the stopping criteria.

Moreover, since two slightly different intelligent approaches are being considered, each approach has to have some parameters initialized. In the case of Q-Learning, for example, each UAV needs to have its action-value function initialized with zeros before beginning to explore the environment, and the exploration and learning rates (ϵ and λ, respectively) need do be defined as well. SARSA on the other hand, because it uses a linear VFA, needs to known a priori which features will be measured and considered in order to estimate the Q-Table. In terms of features, each UAV considers three features: the number of users allocated per UAV; the mean of SINR all users; the UAV total backhaul throughput. These features were chosen because they provide a good balance between coverage and throughput (minimizing users' dissatisfaction), as they represent parameters both from the RAN (number of users allocated per UAV), as well as QoS, such as the mean SINR of users and backhaul throughput. Lastly, it is considered that ground BSs and UAVs exchange information about how many users they have allocated, so that the global reward can be computed by each UAV. Algorithm 5 shows an implementation of the Q-Learning framework, while Algorithm 6 shows the implementation considering SARSA with VFA.

5.5 Numerical Results

5.5.1 Simulation Scenario

Figure 5.2 shows the simulation scenario at different snapshots (episodes) of the network. In this figure, a square urban area is simulated, in which grey squares represent buildings with different altitudes. A three-sectored macro BS (orange

Algorithm 5: Proposed framework - Q-Learning

```
for Every Run do
    Initialize network and user parameters
    for Every episode do
        for Every UAV do
            Initialize UAV current 3D position, s_t
            for Each iteration do
                Choose action a_t, according to π
                Take action (move UAV)
                Get reward r_{t+1} (total users allocated)
                Move to next 3D position, s_{t+1}
                Update action-value funciton according to (2.1)
                s_t ← s_{t+1}
                if Stopping criteria met then
                    Stop UAV
                end
            end
            Communicate reward, r_{t+1}, to other BSs
        end
        Decay exploration rate, ϵ
    end
end
```

triangle) is positioned in the middle, while five small BSs are randomly positioned (coloured squares). Users allocated to any BS are coloured according to that BS's colour and are represented by circles, whereas black Xs represent users out of coverage. From Figs. 5.2b and 5.2c, it is clear that once the users start moving to the event the ground network is incapable of handling this large concentration of users in a specific sector. Also, because both the location as well as the amount of users attending the event is unknown a priori, the deployment of UAVs at predetermined locations is not efficient.

Figure 5.3 shows a snapshot of the network when intelligent UAVs are deployed. In this figure, the UAVs are represented by the coloured diamonds flying above the area, and the path of one UAV for that episode is shown by the dashed line. It can be seen that the UAVs explore the area looking for the position that would yield the best reward, and that in the end, it goes to the best possible position for that episode. In addition, UAVs learn to be in different altitudes, as by changing their height, UAVs are able to provide coverage to a greater area, whereas in lower heights, a smaller area is covered, but a higher SINR can be achieved. Furthermore, UAVs also tend to stay away from both the macro and small BSs, in order to reduce interference and also allocate users that are out of coverage or in crowded areas. Regarding the position between UAVs, it can also

Algorithm 6: Proposed framework - SARSA with VFA

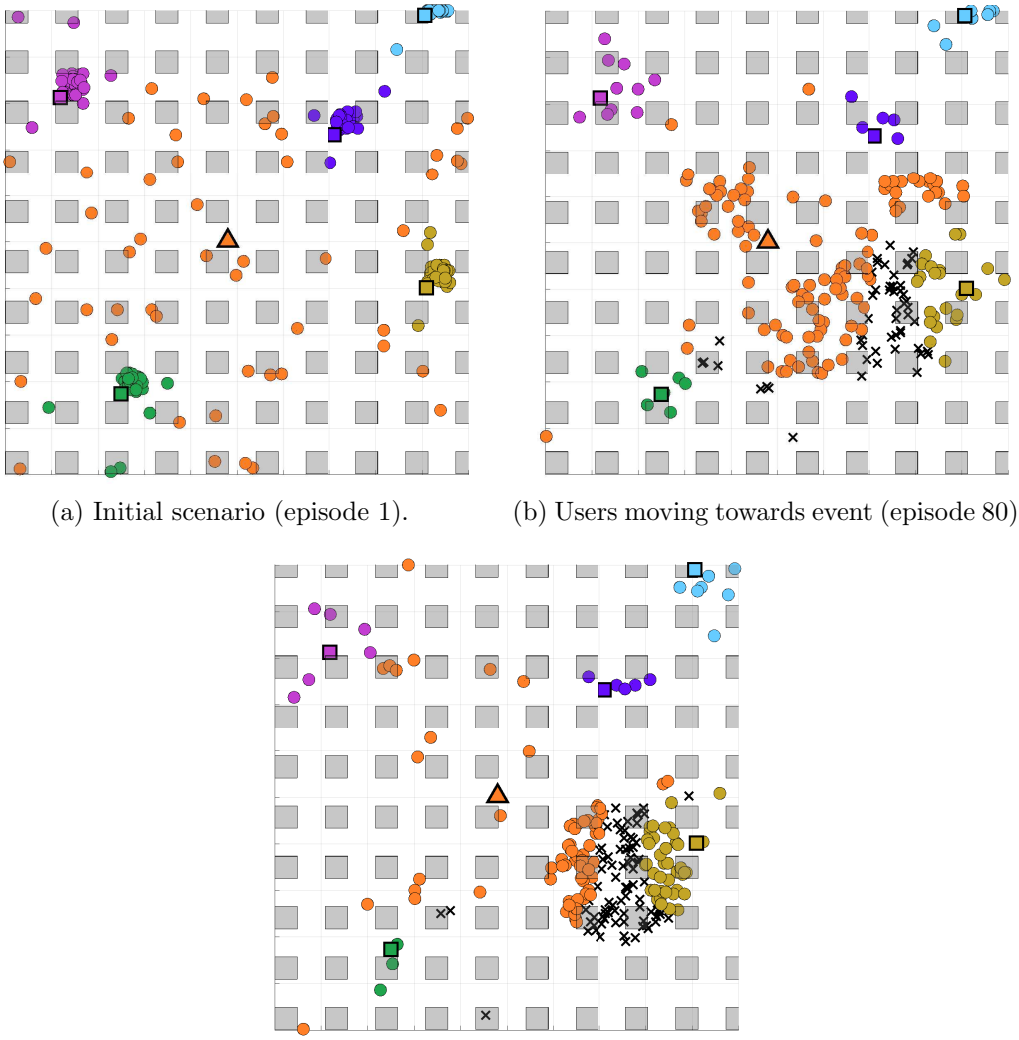
```
for Every Run do
    Initialize network and user parameters
    for Every episode do
        for Every UAV do
            Initialize current 3D position, s_t
            for All actions do
                Get features of current state-action, F
                Estimate action-value funciton according to (5.7)
            end
            Choose action a_t, according to π
            for Each iteration do
                Take action (move UAV)
                Get reward r_{t+1} (total users allocated)
                Move to next 3D position, s_{t+1}
                for All actions do
                    Get features of next state-action, F
                    Estimate action-value funciton according to (5.7)
                end
                Choose next action a_{t+1}, according to π
                Update weights θ⃗ according to (5.8)
                s_t ← s_{t+1}
                a_t ← a_{t+1}
                if Stopping criteria met then
                    Stop UAV
                end
            end
            Communicate reward, r_{t+1}, to other BSs
        end
        Decay exploration rate, ϵ
    end
end
```

be seen that UAVs tend to be spaced out from each other to avoid interference.

5.5.2 Simulation Results

In this section, simulation results for conventional Q-Learning and SARSA with VFA are presented. The intelligent solutions are compared with a scenario where no UAVs are deployed and the metrics considered are:

- Total number of users in outage;
- Macro BS, small cells and UAV average RAN load;
- Average backhaul throughput of the UAVs;



(a) Initial scenario (episode 1).

(b) Users moving towards event (episode 80).

(c) Users at the event (episode 150).

Figure 5.2: Snapshots of the network at different moments. Colored circles represent users associated with the respective BS, while black X's are users in outage.

- Total user dissatisfaction rates for both high and low throughput requirements

User dissatisfaction can be defined as

$$S_{\mathrm{u}} = \frac{T_{u,x} - T_{u,\mathrm{req}}}{T_{u,\mathrm{req}}}, \tag{5.9}$$

where $T_{u,\mathrm{req}}$ corresponds to the required throughput of user u, while $S_{u,\mathrm{H}}$ and $S_{u,\mathrm{L}}$ represent the dissatisfaction rate for users with high and low throughput requirements, respectively.

A three-sectored macro cell, $m = 1$, was positioned in the centre of the area, $k = 10$ small cells were randomly deployed and $u = 600$ users were scattered. The ground network was assumed to have an ideal backhaul, while UAVs are

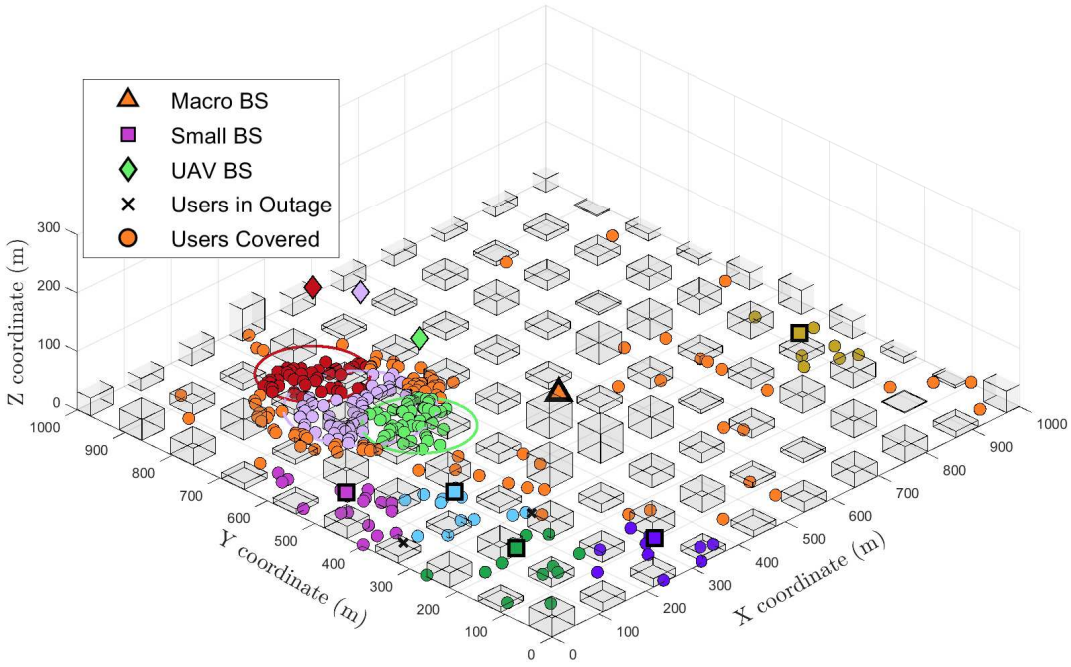


Figure 5.3: Snapshot of the network when UAVs are deployed. The UAVs, coloured diamonds, are seen flying above the event area, providing coverage to users.

assumed to have an out-of-band microwave backhaul with a limited capacity and fixed delay. In addition, a fixed number of $v = 9$ UAVs was evaluated for both conventional Q-Learning and SARSA with VFA. A total of 30 different runs of the simulation were performed, with different small cell and event locations, so that the performance of both RL solutions are averaged out. Table 5.1 shows the simulations parameters.

Figure 5.4 shows the total percentage of users in outage. As it can be seen, initially the network operates with low levels of outage, around 1% to 2%, as expected. However, when the event condition triggers, users start moving to the event location and the outage slowly increases, reaching a peak of about 57%. After a certain threshold is reached (30% of users in outage), the UAVs are deployed, and after that, the outage returns back to its original level, around 1 to 2%, just after around 20 episodes. This emphasizes that UAVs are able to find users out of coverage of both the macro and small cells and connect such users. In addition, since by the time the UAVs are deployed the users are still moving to the event, the behaviour observed in Figure 5.4 tells us that the UAVs are able to track the movements of users in the area from their initial positions through to the event. When comparing the RL solutions, it can be seen that their performance is very similar, as both solutions are able to quickly restore the outage levels back to normal. This works as expected, as both solutions should converge to the same

Table 5.1: Simulation Parameters

Parameters	**Value**
Ratio of build-up to total land area, α	0.3 [221]
Average number of buildings per km^2, β	500 [221]
Scale parameter for building heights, γ	15m [221]
Side of the square area, L	1km
Number of macro BS, m	1
Number of small cells, k	10
Number of UAVs, v	9
Total number of users, u	600
Ratio of users near small cells	2/3
Ratio of users attending the event	$(60\%, 80\%)$
Event radius, R_e	150m
User mobility (slow / fast)	4/10m/s
Ratio of slow/fast users	70/30%
Throughput requirement (low / high)	200/600kbps [2]
Ratio of users with high/low requirement	50/50%
UAV antenna directivity angle, ϕ_{ap}	60° [226]
Carrier frequency, f_c	1GHz
Macro BS height, h_m	20m
Users height, h_u	1.5m
Additional loss (LoS), η_{LoS}	1 dB [223]
Additional loss (NLoS), η_{NLoS}	20 dB [223]
Macro BS EIRP, EIRP$_m$	10dBW
Small Cell EIRP, EIRP$_k$	3dBW
Aerial BS EIRP, EIRP$_v$	3dBW
RB$_{b,\max}$	50 [2]
Bandwidth of 1 RB, B	180kHz
Backhaul overhead factor, ρ	1.3 [2]
$x_{v,\text{step}}, y_{v,\text{step}}$	50m
$z_{v,\text{step}}$	100m
$C_{v,\max}$	100Mbps [2]
UAV heights	$(200, 1000)$m
Total number of episodes	200
Number of independent runs	30
Max iterations per episode	$1,000$
Max iterations, same reward	100
Learning Rate, λ	0.5
Discount Factor, ψ	0.9
Gradient Descent learning rate, λ_{GD}	1e-6
Exploration rate, ϵ	{0.8,...,0.1}

learned parameters, as they utilize the same reward metric. The great advantage of VFA, however, is the needlessness of the lookup tables, which allows the UAVs to be more independent and to adapt themselves to any environment. Because now the action-value function is replaced by a linear function, it also reduces the memory requirement, as the UAVs only need to remember the learned weights, $\vec{\theta}$, instead of a table with all possible states and actions.

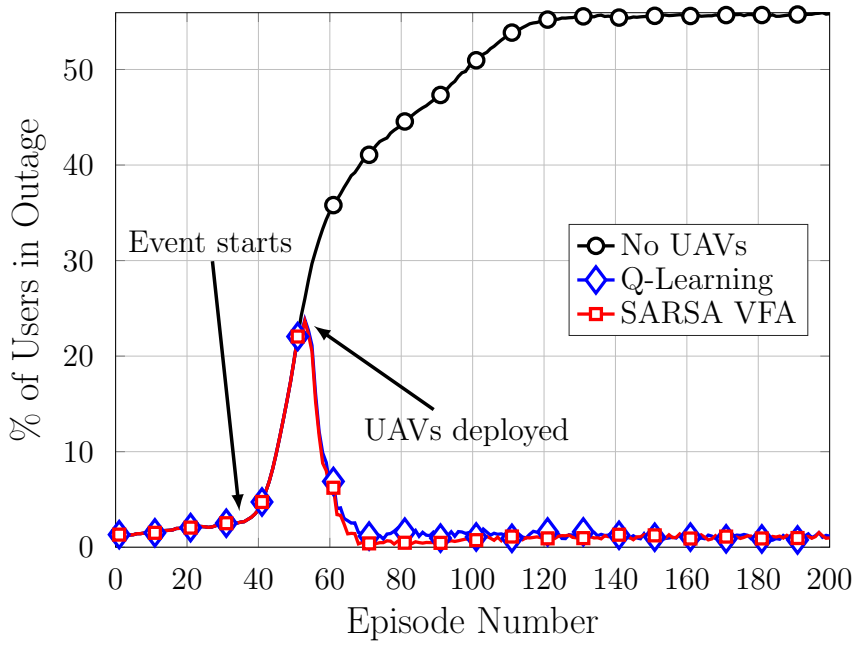
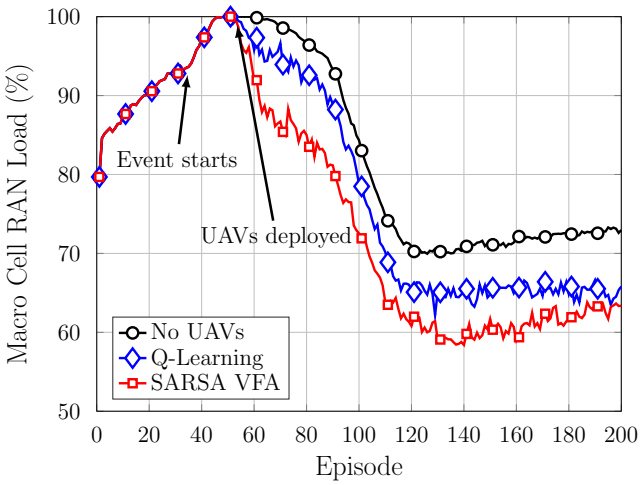


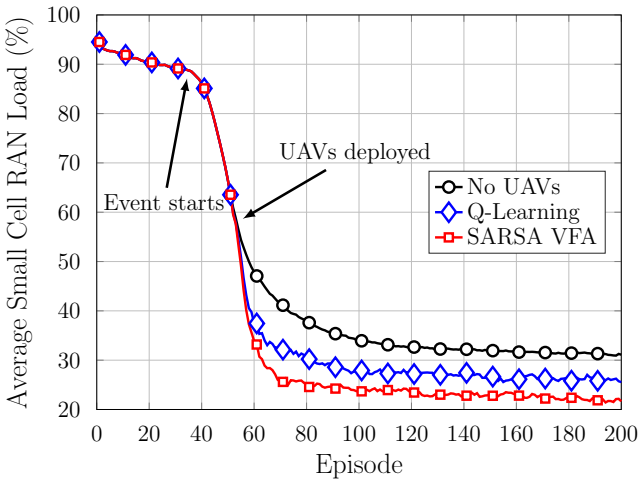
Figure 5.4: Percentage of users in outage.

Figures 5.5 (a)—(c) show the average macro cell, small cells and UAV RAN load, respectively. When the event happens, users start moving away from the small cells towards the event, causing the macro cell to be completely loaded, and users to be in outage. After this brief overload period the load of the macro cell starts to drop, as more and more users are going to the sector that the event is happening, increasing the number of users in outage, leading to the result in Figure 5.4. When comparing the RL solutions with the no UAV scenario, the macro and small cells loads do not change much. This works as expected, as the UAVs are more concerned in providing coverage to users out of coverage, rather than "stealing" users that are already allocated to a ground BSs. In addition, the average UAV RAN load is around 80% for Q-Learning and 87% for SARSA with VFA, meaning that the UAVs are performing well in terms of user allocation. Furthermore, it can also be seen that SARSA with VFA has better average UAV RAN load. This can be explained due to the features chosen by the VFA algorithm, which also attempts to prioritize user satisfaction. Thus, as it can be seen from Figure 5.5, SARSA with VFA does "steal" some users from both the macro cell as well as small cells, in order to provide them a better service. On the other hand, because conventional Q-Learning is only concerned about allocating users, the same behaviour cannot be observed.

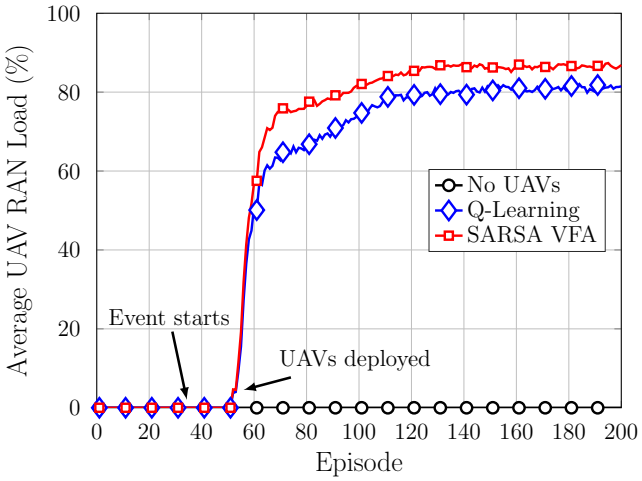
Figures 5.6a and 5.6b, show the total average percentage of dissatisfaction per user for users that have high and low requirements of throughput. In terms of dissatisfaction, it can be seen that whenever UAVs are deployed in the net-



(a) RAN load of the macro cell.



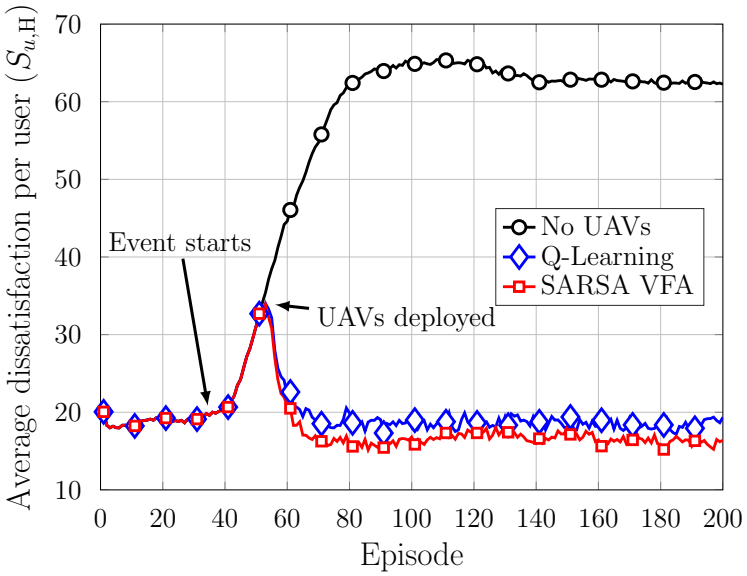
(b) Average small cell RAN load.



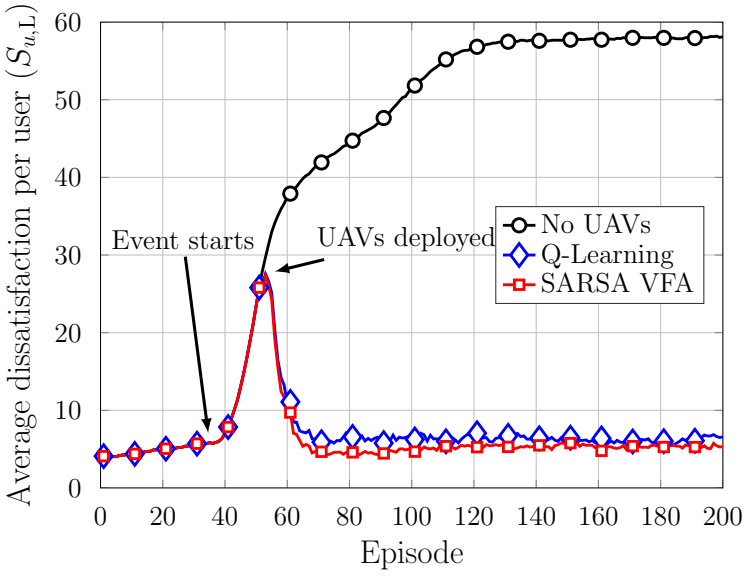
(c) Average UAV RAN load.

Figure 5.5: Percentage of RAN load of the macro cell, small cells and UAVs.

work both users with high and low requirements have their average dissatisfaction rates mitigated. In addition, SARSA with VFA is capable of reducing the av-



(a) Average dissatisfaction per user for users with high throughput requirement.



(b) Average dissatisfaction per user for users with low throughput requirement.

Figure 5.6: Average dissatisfaction per user, in %, for users with high and low throughput requirements.

erage dissatisfaction when compared to conventional Q-Learning. This happens because the features selected also prioritize the mean SINR of users allocated, thus, another advantage of VFA can be observed. Not only VFA is able to reduce the memory needed by conventional RL algorithms, but it also enables them to add supplementary metrics to the objective function (reward), making the system learn to prioritize other parameters as well. In other words, VFA can deal with multiple features with reasonable complexity, while conventional RL cannot.

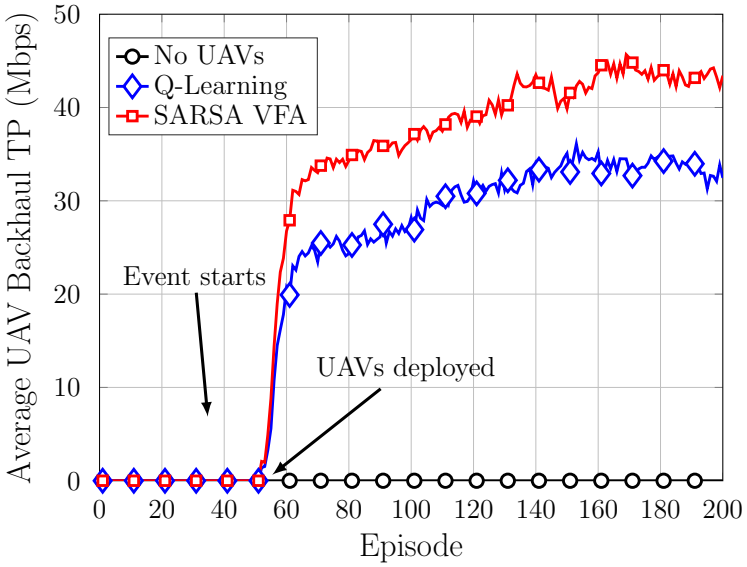


Figure 5.7: Average backhaul throughput of the UAVs.

Lastly, Figure 5.7 shows the average backhaul throughput of UAVs. As it can be seen, SARSA with VFA outperforms conventional Q-Learning by about 40%. This occurs because in VFA the backhaul throughput of UAVs was considered as a feature, which allowed the algorithm to prioritize not only allocating users, but also occupying more efficiently the backhaul.

5.6 Complexity and Convergence Analysis

In order to evaluate the complexity and convergence of the RL solutions, a small simulation scenario is developed. This is performed because it is easier to prove the convergence of the algorithms at a smaller scale, and to evaluate and compare the complexity of RL approaches with an exhaustive search method. The reason for this is to demonstrate that RL solutions are able to find the optimal position without any knowledge of the environment.

In this scenario, $u = 20$ static users are placed in a fixed configuration and the ability of a single UAV ($v = 1$) to find the best possible path and 3D position in this grid is evaluated. The grid consists of a 10 by 10 by 5 m^3 area, with steps of 0.5 metre (a total of 2,000 positions), with the UAV being able to position at a minimum and maximum height of 10 and 12 metres, respectively. Both Q-Learning and SARSA with linear VFA are tested and the same framework as described in Sub-Section 5.4.1 is considered. The intelligent solutions are compared to an exhaustive search approach. Figure 5.8 shows the small scenario, where the UAV, represented by a blue diamond, with its coverage radius shown

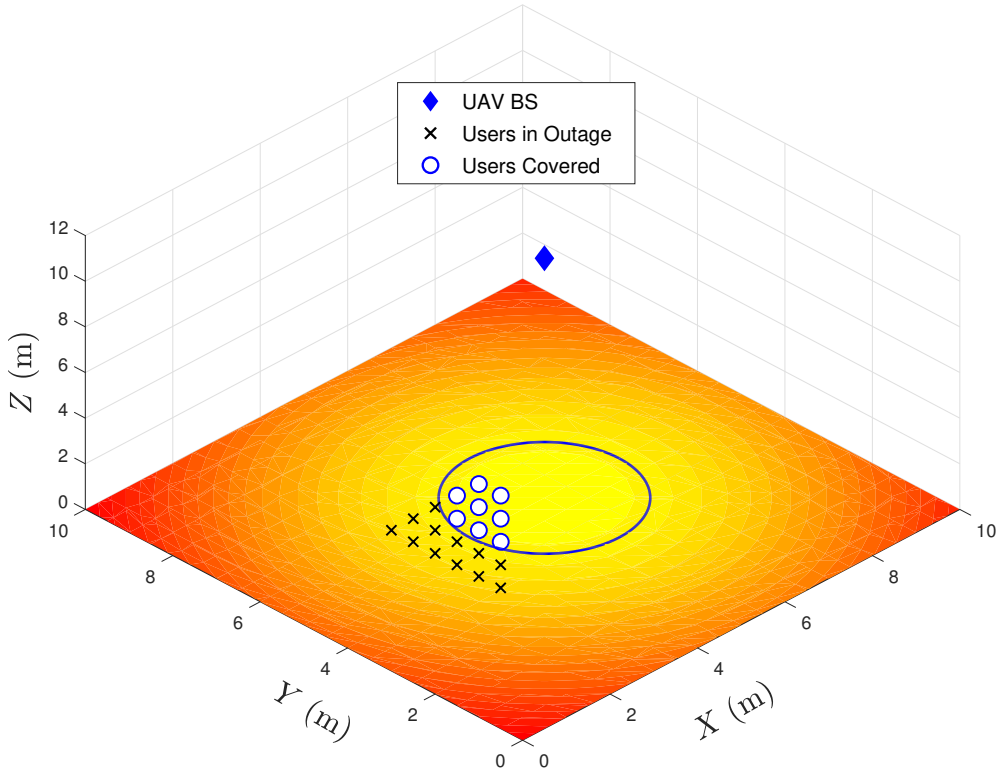


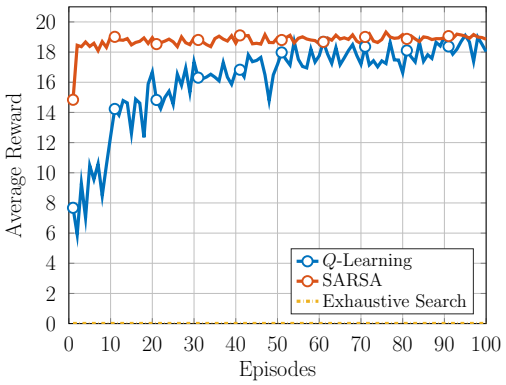
Figure 5.8: Isometric view of the small simulation scenario.

as a blue circle, flies around and learns autonomously the best positions and path towards the users. White circles represent users covered by the UAV, black Xs represent users out of coverage and a heat-map of the RSRP strength at ground level is also shown.

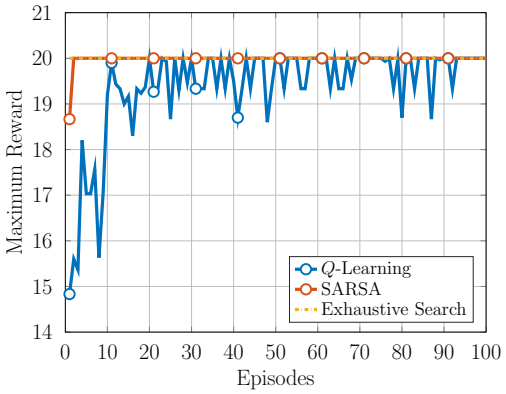
In terms of simulation parameters, 100 episodes are considered. An episode is defined as the moment that the agent is deployed until it gets to the point of maximum reward (in this case, when all users are covered). In addition, in each episode the UAV could perform up to 2,000 iterations (stopping earlier if it reaches the goal). For both Q-Learning and SARSA, with $\lambda = 0.8$, $\phi = 0.9$ and $\epsilon = 0.8$, with ϵ decaying by 0.05 every 20 episodes (until it reaches a value of 0.1). In terms of VFA, a GD learning rate of $\lambda_{\text{GD}} =$ 1e-4 is considered and three different features $n_f = 3$ are chosen, in order to achieve a balance between allocating users and providing them with a better RSRP. The features are:

- Total number of users allocated to UAV;
- Sum of RSRP of all users;
- Mean RSRP of all users.

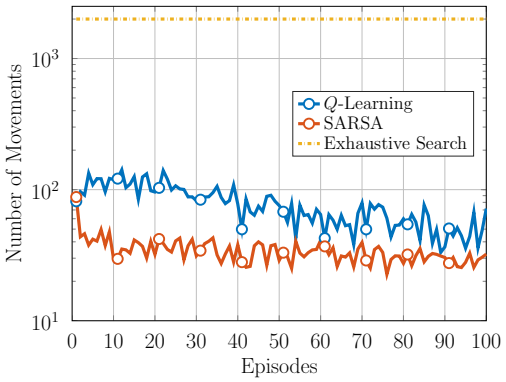
For the small scenario, the UAV starts in a random position every episode (after it has reached the goal, the UAV's position is reset to a random state) and



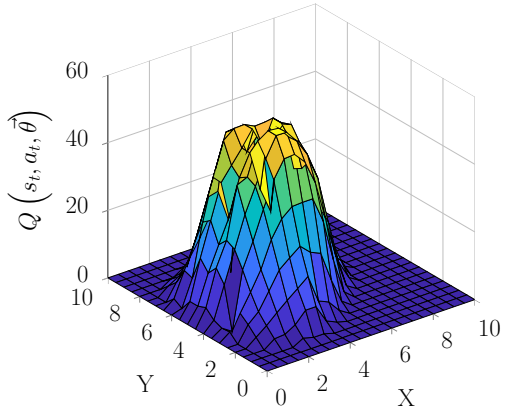
(a) Average reward per episode.



(b) Maximum reward per episode.



(c) Number of movements to reach maximum reward.



(d) Learned $Q(s_t, a_t, \vec{\theta})$ by VFA for $z_v = 10$.

Figure 5.9: Small scenario showing the performance of Q-Learning and SARSA when compared to an exhaustive search method.

a total of 30 independent Monte-Carlo runs are performed, so that the results can be averaged out. Both Q-Learning and SARSA with VFA are compared with an exhaustive search approach, and the results can be seen in Figs. 5.9a, 5.9b, and 5.9c. Figure 5.9a shows the average reward (number of users covered by the UAV) per episode, which is calculated by summing all the cumulated rewards of one episode and dividing it by the total number of iterations that the UAV performed in that episode. As it can be seen, the exhaustive search needs to inspect all possible positions (all 2,000 values) in order to determine the best placement of the UAV, thus its average reward is very low (near zero). Q-Learning, on the other hand, starts with a relatively low reward (because it has not explored the environment and filled the Q-Table enough) and later on converges to a value near the maximum reward of 20. Contrarily, SARSA with VFA learns much quicker and converges much faster. This is due to the representation of the Q-Table by features, which allows VFA to generalize across different states and actions, as well as learn the best parameters to optimize.

Figure 5.9b shows the maximum reward per episode. As it can be seen, the exhaustive search always achieves the maximum reward, but this comes at the price of having to evaluate all possible positions at every episode. On the other hand, Q-Learning starts with a relative low value of maximum reward, but after around 30 episodes it converges to the maximum and oscillates between 19 and 20. Lastly, SARSA with VFA performs really well, starting with a value below 19 and having the same performance of the exhaustive search for the remaining episodes. Figure 5.9c shows the minimum number of movements necessary for the UAV to reach the maximum reward. As expected, the exhaustive search needs the most movements, while the intelligent methods outperform it by a large margin. It can be seen that Q-Learning converges to around 60 movements per episode and SARSA to around 30 movements per episode. This also highlights the advantages and huge potential of VFA, as it is able to learn much faster, due to the inclusion of features in the model, being able to learn better paths and actions to take reducing the number of UAV movements. In addition, despite not being considered in this work, this also shows the potential advantages of using RL in terms of energy consumption of UAVs, as intelligent methods are much more efficient than exhaustive search. Lastly, Figure 5.9d shows the learned action-value function $Q(s_t, a_t, \vec{\theta})$ by SARSA. As it can be seen, SARSA learns that being near users has a better function value, and being away leads to worse values. Furthermore, it can also be seen that the peak of the action-value function occurs when the number of users covered is the maximum. It can be concluded that both intelligent methods, Q-Learning and SARSA, achieve the optimal solution with a considerable lower number of movements, indicating that RL is a suitable strategy for this problem. Lastly, it can also be seen that RL is highly suitable and desirable to perform model-free tasks, in which previous knowledge about the network, such as knowing the users positions or requirements is not necessary. As such, RL approaches can achieve optimal solutions requiring almost no previous information, unlike other methods.

5.7 Summary

UAVs are expected to play a crucial role in future mobile networks, enabling many applications that are unimaginable today. However, for that to be possible, an intelligent deployment of UAVs is essential, in order to guarantee the aerial network both adaptability and autonomy. As such, a use-case of intelligent UAVs deployed in the context of pop-up mobile networks was envisioned and two different algorithms (based on Q-Learning and SARSA) were proposed to

determine the optimal 3D positions of multiple UAVs.

In addition, simulations in a pop-up network scenario were performed, in order to test the scalability of the proposed framework. It was shown that both Q-Learning and SARSA are able to determine the optimal trajectories of UAVs and reduce the outage levels of the network back to its regular performance while also minimizing user dissatisfaction rates. In addition, SARSA with VFA slightly outperformed conventional Q-Learning due to its inclusion of features in the model, which make RL solutions more robust and generic. After that, experiments in a small simulation scenario were conducted, in which it was shown that approaches based on RL can converge to optimal solutions with limited complexity and without any previous knowledge about the network or users positions. This clearly shows the suitability of intelligent approaches for this kind of problem.

Chapter 6

Conclusions, Future Trends and Open Issues

In this section conclusions are drawn in terms of SON, as well as for each use-case investigated in this thesis. In addition, an overview on future trends expected to be seen in RL is presented. This covers trends in both conventional RL, as well as the more recent field of deep RL, which has gained increased attention in the recent years. After that, an overview of future trends expected in SON is presented, covering general ML techniques. However, since the focus of this thesis is the application of RL in SON, a summary of future trends in RL applied in SON is also presented. Last, some potential future research directions are identified, with the focus on the use-cases investigated in this thesis.

6.1 Conclusion

In order for future networks to keep updated and on par with state-of-the-art intelligent systems a change in paradigm needs to be developed and this will most likely require the use of intelligent solutions, mainly ML algorithms. Future networks will also require a change in the way the network is perceived. In the future, thousands of parameters will need to be configured, thousands of cells will need to be monitored and optimized at the same time and a huge amount of data will be collected, not only from humans, but also from machines. Since it is impossible for humans to deal with this amount of tasks and data, ML solutions will need to be applied in order to learn models in a relative short amount of time and to enable an autonomous and intelligent network. As such, ML is expected to play a vital role in future mobile networks in order to enable them to achieve their full potential. In this context, RL solutions are considered to be highly suitable for future mobile network scenarios, specially because of their ability to

learn in real time and with their interaction with the environment. Furthermore, deep RL with its inherit ability to learn directly from raw data complex patterns will also play an important role in future mobile networks, by enabling the design of autonomous systems with a higher level of understanding.

Based on this, this thesis covers some applications of RL algorithms in self-optimization use-cases of SON. First, a literature review covering the areas of ML applied in SON was presented. In this review, the three main areas of SON were presented (self-configuration, self-optimisation and self-healing), and its use-cases were presented. In the context of self-optimisation, the focus of this thesis, an analysis of how different ML techniques were applied in self-optimisation use-cases was presented, together with which problem they aimed to tackle. After that, the use-case of user-cell association involving the optimisation of network and user parameters was investigated. In future mobile networks, it is expected that the backhaul will become the bottleneck, instead of the RAN, as such, current and future mobile networks are in need of cell association procedures that consider end-to-end connectivity and backhaul constraints. Based on that, a two-step Q-Learning solution was proposed, in order to, first optimize SC parameters (CREO), and then, after the CREOs are communicated to users and depending on individual user requirements, optimise the user weights. Results have shown that our proposed approach, user-specific cell association, outperforms three other state-of-the-art solutions, highlighting that optimising both network and user parameters yield better results than only network ones. This enhanced performance in terms of user QoS parameters, such as throughput, latency and reliability, however, comes at the expense of a minor reduction in overall throughput. However, as the results have indicated, the proposed solution is flexible enough in order to find a balanced trade-off between these metrics, as well as to adapt to changes in the network, while still converging in a relatively quick manner.

After that, this thesis investigated the use-case of coverage and capacity optimization in SON, through the utilization of intelligent UAVs coupled with different RL solutions. First, an emergency communication scenario was presented, in which UAVs were utilized as access points in order for isolated people, as well as search and rescue teams, to communicate to each other. In this scenario, the problem of determining the optimal 3D placement of multiple UAVs based on user requirements and network constraints was considered. As such, a distributed solution based on Q-Learning was proposed and it was compared to three other schemes. Results have shown that given enough time the UAV BSs were capable of learning user mobility and to track users that were not associated to any BS, as well as to adapt to different user mobility patterns and requirements. Fur-

thermore, when comparing the the proposed approach to the other schemes, the Q-Learning proposed solution achieved a better performance in terms of all considered metrics, mainly: users covered, user throughput satisfaction, as well as better network resource utilisation (in terms of the backhaul).

Lastly, the deployment of UAVs was also investigated in order to provide additional capacity to overloaded mobile networks, more specifically in the context of pop-up networks, such as when big events happen. In this context the problem of optimising the 3D positioning of multiple UAVs considering different user requirements, mobility, as well as interference from the ground network and both ground and flying network constraints was studied. Due to the complexity of the proposed network model, in this problem two intelligent approaches were investigated, one, based on conventional RL, Q-Learning, and another one, based on VFA, SARSA. Results have shown that both proposed approaches were able to restore the network to its previous levels of users connected and user satisfaction rate, with SARSA having a slightly better performance than Q-Learning. This occurs mainly due to the problems that conventional RL has when the state space becomes too large, since traditional RL algorithms encounter issues in terms of memory and generalization to new unseen states. As such, the gains presented by SARSA translate not only in terms of performance, but also regarding memory and generalization, due to its combination with VFA, which allows the table to be represented by any function.

6.2 Future Trends in RL

6.2.1 Conventional RL

Reward Design

One challenge that has attracted the attention of researchers in the field of RL for many years is the problem of reward design. More specifically, how to create an objective function that represents the problem accurately enough and that will guarantee that the agent is capable of producing desirable results, while avoiding undesired solutions [10, 247]. As such, the design of the reward signal is crucial for RL systems, as the reward is the only feedback signal that the agent has on how well it is performing in the environment, thus, a tiny change in the reward function can sometimes lead to completely different solutions [10].

Transfer Learning

Transfer learning is a new research area that has emerged in the context of RL which focuses on the development of algorithms that are capable of transferring the knowledge from an agent which was trained on certain source tasks to a target task [248]. The idea is to transfer knowledge between similar tasks so that the new learning agent can use previous knowledge of the source tasks in order to solve the target task in a more quick and effective manner, by for example, reducing the number of steps needed for it to achieve a near optimal performance. This would be extremely useful, because current RL algorithms need a lot of samples to learn a near optimal solutions, which, sometimes, can be prohibitive, such as in cases of real world applications. As such, transfer algorithms should be able to build prior knowledge from different, but related tasks and improve the performance of the RL agent [10,248]. However, despite these benefits, several challenges still remain in the transfer learning domain, such as which parameters should be transferred (which parameters are relevant in both domains, for example), and also in terms of theoretical proofs [248].

Exploration-Exploitation Trade-Off

Another common challenge in RL is the trade-off in exploration and exploitation, more specifically in how to handle the exploration of an agent. In complex tasks, for example, in which rewards are sparsely available, it can be nearly impossible for an agent to learn, without any prior knowledge, the considered task (as it would be extremely rare for the agent to discover which states and actions lead to rewards) [248]. As such, concepts like imitation learning have started to emerge in RL, in which the agent is, first, presented with some examples of good policies and behaviours, so that the agent can learn and try to imitate what was previously observed.

Function Approximators

Lastly, another topic that has seen some discussion in recent years is which function approximators to use in RL tasks. Traditionally, RL algorithms have used a table in order to represent its states or state-action pairs, however, as previously seen, for these algorithms to be able to scale up to bigger and more complex tasks and to be able to generalize, function approximation techniques are needed [10]. In order to represent the value functions any regression algorithm from the ML field can be used, but one question that has intrigued researchers for quite some time was for what type of problem which function approximation is the best

one [248]. Recently, however, the research community seems to have converged to a common approximator, with the development of a new field in RL, deep RL, in which deep NNs are utilized as functions to represent the value function of RL algorithms [34, 51–54]. However, despite this common denominator, still several trends and open problems are present in deep RL, as it will be seen next.

6.2.2 Deep RL

Recent advances in deep learning, more specifically in deep NNs, in which several layers are utilized to build abstract representations of raw input data, have made it possible for these NNs to learn complex concepts, such as categorizing objects, recognising human speech or diagnosing diseases [34]. As such, it was only natural that these algorithms were combined into RL, in what has become, recently, the new field of deep RL. Although several breakthroughs have been achieved by deep RL in recent years, several areas still need to be explored and open issues still remain, as it will be seen next.

Although Deep RL also suffers from some of the problems encountered in conventional RL, such as the exploration-exploitation trade-off, the amount of samples needed to solve a certain problem, how to design a meaningful reward for complex tasks and also how to transfer knowledge across different domains, deep RL also has its own problems due to the differences in the algorithms structure [247, 249].

Deep Neural Network Architecture Design

As seen in recent years, deep NNs have become a powerful tool as it enables algorithms to learn directly from its raw input data, requiring almost no human intervention. However, these type of NNs, due to their size and complexity can have various parameters that need to be tuned and considered in its design in order to achieve its goal. These parameters can be numerous, and consist of basic design parameters, such as the number of network layers, the number of hidden neurons in each layer, learning rate, dropout rate, the activation function that will be used, or the initial weight initialization, to name a few [33].

As such, in order to design a deep RL problem, the same issues are encountered, in which the NN parameters must be carefully tuned in order to achieve a near optimal solution for a specific task. Moreover, the parameters of the deep NN must be even more carefully tuned in the case of deep RL, as most conventional algorithms do not support non-linear function approximators. This occurs due to the correlations present in the sequences of observations performed by the

agent, as well as the correlation between action-values and target values, and also the fact that small changes in the value function can significantly change the policy being followed [34].

Imperfect Information Environments

Another trend that has seen increased attention in deep RL is the concept of imperfect information games. Many real world applications can actually be modelled as a game of imperfect information, in which an agent only knows a portion of the state of the game and must infer the the other portions based on the information of other players actions, such as in a card game, where each player's card are hidden from the rest of the players. [250]. Although several ML algorithms have achieved near optimal solutions to classical, perfect information games, these algorithms fail to converge when imperfect information environments are considered [250].

Although solutions to imperfect games would be a Nash equilibrium, methods that attempt to find it lack the ability to learn intrinsic patterns and generalize for new situations, or in other words, they lack generalization capabilities. As such, solutions that attempt to find the Nash equilibrium analytically are limited by the complexity of the problem. Thus, similar to the case of VFA, solutions that are capable of abstracting information, providing generalization across different conditions, such as deep RL, are preferred [250].

6.3 Future Trends in SON

In order for future mobile networks to overcome current network limitations, it is clear that a shift in paradigms is needed and that different solutions to common problems need to be found. However, despite current work being done in the area of SON, with an increase of maturity and robustness in the area, with more and more different ML algorithms being explored and applied in different contexts, there are still open issues and challenges that need to be addressed in order to enable a fully intelligent network in the near future. As such, in the next paragraphs a brief overview of future trends and open issues in the context of ML applied in SON is presented.

6.3.1 Machine Learning in SON

In order for the concept of SON to be fully enabled, it is clear that more intelligence needs to be added to mobile networks. In addition, with the already

extensive collection and monitoring of network data by network operators, ML algorithms have truly an unimaginable potential to shine in future mobile networks, enabling applications and services that are unimaginable today. However, despite this fact, not many ML solutions have been explored in the realm of SON. Although not every algorithm is recommended to be applied to every *self-x* function, further exploration of ML solutions still need to be done in order to investigate their performance and determine if these methods can really work or not.

On top of that ML algorithms really have a great opportunity to excel in future mobile networks, as the amount of data collected and generated by future networks are going to be enormous, providing virtually an infinite dataset for algorithms to train. However, depending on the type of data collected and the necessary learning, different types of ML solutions are more suitable for different problems. Below, a brief overview on how supervised, unsupervised learning, and deep learning can play an essential role in future networks is presented.

Supervised Learning

As previously mentioned, supervised learning requires information both from input and output data in order to learn a model and make predictions about the future. Due to their natural characteristic of learning based on the output feedback, supervised learning algorithms are limited in the amount of applications that they can cover in future mobile networks. This occurs because it can be extremely hard for operators to generate labelled data for certain network use-cases. For example, in the case of self-healing, operators might not have in real-time information about complete network status, or where exactly the failure happened. Thus, in those cases other ML algorithms are more suitable to solve these problems. On the other hand, supervised learning can excel at other use-cases for future mobile networks, where labelled data is available and real-time information is not necessary. Some future envisioned applications for supervised learning could be in the area of mobility management, resource allocation, self-healing (in terms of fault detection and classification), etc. However, despite all of these applications, supervised learning still has to be further explored in different contexts of future mobile networks.

Deep Learning: moreover, one area that has seen a lot of growth in recent years is the concept of deep learning. Deep learning has already proved to to be really powerful algorithms which were able to improve state-of-the-art solutions in speech recognition, object detection and genomics, for example [33]. In the

context of mobile networks, deep learning has seen an increased attention in recent years, more notably in its application at the physical layer level [251, 252], security [253, 254], mm-Wave communications [46, 255], and resource allocation [256, 257]. As such, due to its ability to learn features directly from raw input data and its performance above human level, deep learning is expected to play an important role in future mobile networks and to solve an even larger number of applications and use-cases.

Unsupervised Learning

Unsupervised learning on the other hand is able to deal with unlabelled data very well. As such, unsupervised learning algorithms are expected to play a vital role, in grouping or clustering applications, such as in fault management [258–261] or resource optimization [155, 161], for example. However, similarly to supervised learning, unsupervised techniques still has to be further considered in different network scenarios, such as in scenarios that consider massive MIMO, ultradensification of the network, the new network structure of 5G new radio, or even M2M communications, such as in the case of V2X.

6.3.2 RL in SON

In addition to the other ML techniques, one technique that can play an extremely important role in future mobile networks is RL. Due to their inherit nature of being a goal-oriented approach, which interacts with the environment, generates samples from it and learn from previous conditions, RL algorithms are capable of online learning without any human intervention. As such, these algorithms are ideally suited to solve dynamic problems in a heterogeneous and changing environment, such as mobile networks, specially in the use-cases of self-optimization and self-healing, as they are able to constantly sample the network and determine the best actions to take online. However, despite RL being extensively applied in several other domains, such as in computer science or gaming [10], RL has still not seen a great number of applications in mobile networks.

However, it is envisioned that RL will be a crucial part in future mobile networks, due to their innate ability to learn online and by constant interaction with the environment, which is they key difference from other conventional ML techniques. As such, RL solutions, specially those based on value function approximation and deep RL, are expected to see an increased number of applications in future mobile networks.

6.3.3 Deep RL in SON

Deep RL is quite a recent area which has emerged and gained popularity recently, due to their powerful ability to combine deep learning methods with RL. This, by its turn, is able to generate really powerful algorithms that are able to learn and extract information directly from raw input data, and convert into actions that can be taken by an agent in the real world. As such, deep learning enables RL to scale to decision-making problems that were previously considered impossible, due to the their complexity or the high number of possible states and actions. However, despite the recent *boom* in deep RL, specially in the are of computer science, not many issues in mobile networks have seen its application. Some applications that have seen the utilization of deep RL consist of resource allocation [262, 263], interference management [**?**, 246], and content caching [264, 265].

However, because deep learning is enabling RL to tackle problems that were unimaginable a couple of year ago, deep RL is considered to revolutionize the filed of artificial intelligence, by building agents that are able to develop a higher understanding of real world applications with basic to almost none previous information, albeit at the expense of a lot of data. As such, because future mobile networks are expected to generate a huge amount of data every day, deep RL can play a vital role in developing online solutions.

6.3.4 Backhaul Optimisation

As previously seen in Table 2.1, the issue of backhaul optimisation is not addressed enough by the research community, despite the important impact that it can have in future wireless communications, as it is deemed to be the bottleneck of next generation networks. As such, if the backhaul issue is not addressed, specially if the network densification occurs, the expected targets of latency, reliability and capacity for future generations may not be achieved.

In light of that, several optimisation procedures of the backhaul can be considered as extensions of the framework proposed in Chapter 3. One possible extension of the current scenario could be the investigation of a similar scheme, but considering SCs with multiple backhauls with different characteristics. The system could then learn either to choose the best backhaul for each situation or to connect different users to different types of backhauls in order to maximize not only user QoS requirements, but also EE of the network. Another possible future work can consider the case when different contents are being requested by different users. In this scenario, a caching optimization could also be included, in which SCs with contrasting backhaul solutions would learn to cache different contents

depending on their demands. In this situation, then SCs that have backhauls with poor latency conditions could then learn to cache more popular contents in order to minimize the latency experienced by users, whereas SCs with better latency constraints could cache less popular contents, in order to serve specific users.

6.3.5 Intelligent UAVs in Future Mobile Networks

UAVs are expected to be the next breakthrough in wireless networks, mainly due to their flexibility, agility and LoS communication capabilities. As such, the integration of UAVs in mobile network environments has seen a great number of applications in recent years. However, despite several research efforts focusing in the area, several issues still remain in the topic, as highlighted in Chapters 4 and 5, such as finding the optimal trajectory and placement of UAVs, how to manage the interference from the ground network in UAVs and vice-versa, how to perform resource allocation between aerial and ground networks, and how to connect UAVs to the core network (backhaul), to name a few.

Since this is a relatively new area, the 3D placement optimization problem of UAV-BSs constrained by network requirements can have several possible extensions. One extension of the works proposed in Chapters 4 and 5 can be the consideration of other more robust RL algorithms, such as the ones based on deep RL. This would enable UAVs to learn better policies and more complex and intrinsic patterns in the environment. In addition, the consideration of other deep RL frameworks would also allow a better learning model to be built, which eventually can lead to the autonomous deployment of aerial BSs in real world applications.

Another interesting possibility of future work is to consider additional requirements from the network, such as users having demands in terms of latency and resiliency as well. Then, in such cases, optimization would have to be performed considering not only on the connectivity of each user, but also with respect to their satisfaction levels regarding each parameter. Another future topic of research consists of including additional constraints in the UAV-BSs, such as the flight time and the total energy consumption from the UAV-BS's movements. This would enable the algorithm to find a trade-off between moving and coverage, in order to maximize both the amount of users and the flight time of each UAV-BS. Lastly, another topic that can be considered is a non-ideal backhaul link between UAVs and ground BSs, with losses depending on the distance for example. As such, similar to the previous case, a similar trade-off could arise, in terms of distance between the ground BS that provides the backhaul link to the

UAV and the satisfaction level of end-users connected to the aerial platform.

Appendix A

Example Comparing Q-Learning and SARSA

One very common and academic example that can be found in the literature in order to illustrate the differences between off-policy (Q-Learning) and on-policy (SARSA) is the gridworld task of cliff walking [10]. In this example, a world consisting of tiles (or grids) is generated. An agent is inserted in the environment and is able to select between four possible actions, mainly: up, down, left or right. Moreover, in this scenario, it is considered that an agent follows a fixed behaviour policy, ϵ-greedy, with a constant value of $\epsilon = 0.1$ (10% chance of choosing a random action). In addition, specific starting and goal states are set and, the objective of the agent is to reach the goal in the minimum number of movements. The reward of every tile, with the exception of the *cliff* tiles are equal to -1, while the *cliff* region has a reward of -100 and also sends the agent instantly back to its starting position. The top part of Fig. A.1 shows the considered example scenario, adapted from [10], in which both Q-Learning and SARSA agents are tested, while the results obtained are shown below.

As it can be seen, the off-policy learning algorithm, Q-Learning, is able to learn values for the optimal policy (which says that the agent should travel on the edge of the cliff). However, due to the randomness in selecting actions, this also results in the unfortunate events of the agent falling of the cliff occasionally. On the other hand, SARSA takes the action selection into account (due to its on-policy behaviour) and, as such, it is able to learn a longer, albeit safer path towards the goal.

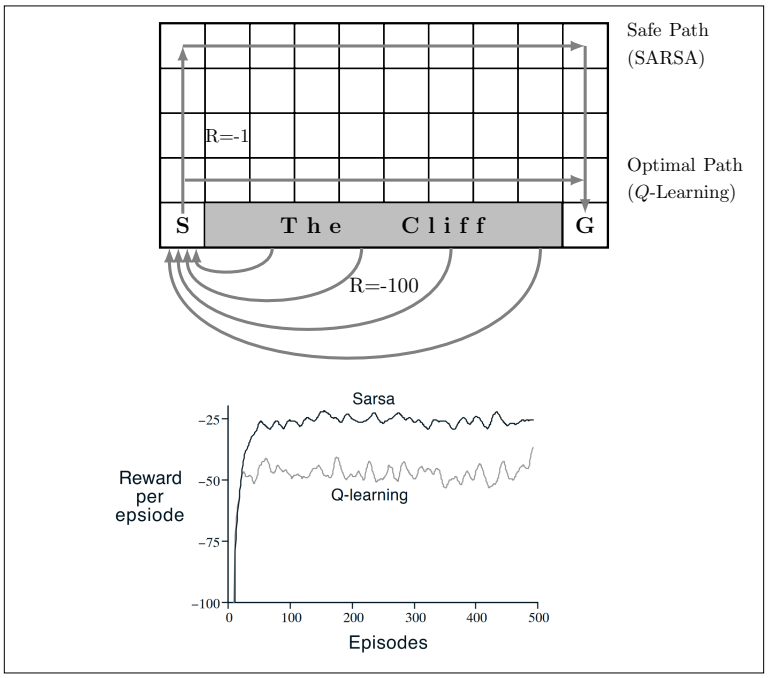


Figure A.1: The cliff walking task, adapted from [10]. As it can be seen, *Q*-Learning learns the optimal path, however, due to random actions, its reward is lower. On the other hand, SARSA learns a safer path, which accounts for the randomness in choosing actions, achieving a higher reward.

Bibliography

[1] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "A multiple attribute user-centric backhaul provisioning scheme using distributed SON," in *Proc. IEEE Global Communications Conf. (GLOBECOM)*, Dec. 2016, pp. 1–6.

[2] ——, "A distributed SON-based user-centric backhaul provisioning scheme," *IEEE Access*, vol. 4, pp. 2314–2330, 2016.

[3] M. Jaber, M. Imran, R. Tafazolli, and A. Tukmanov, "An adaptive backhaul-aware cell range extension approach," in *Proc. IEEE International Conference on Communication Workshop (ICCW)*, Jun. 2015, pp. 74–79.

[4] NTT DoCoMo, "R1-103264: Performance of eICIC with control channel coverage limitation," *3GPP TSG RAN WG1 Meeting*, vol. 61, (Montreal, Canada), May 2010.

[5] NGMN Alliance, "Small cell backhaul requirements," *white paper*, June 2012.

[6] "TR 36.814: Further advancements for E-UTRA physical layer aspects (release 9)," *3rd Generation Partnership Project (3GPP) Technical Specification Group Radio Access Network*, vol. 9.2.0, March 2017.

[7] E. Almeida, A. M. Cavalcante, R. C. Paiva, F. S. Chaves, F. M. Abinader, R. D. Vieira, S. Choudhury, E. Tuomaala, and K. Doppler, "Enabling lte/wifi coexistence by lte blank subframe allocation," in *2013 IEEE International Conference on Communications (ICC)*. IEEE, 2013, pp. 5083–5088.

[8] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[9] P. V. Klaine, O. Onireti, R. D. Souza, and M. A. Imran, "The role and applications of machine learning in future self-organizing cellular networks," in *Next-Generation Wireless Networks Meet Advanced Machine Learning Applications*. IGI Global, 2019, pp. 1–23.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An introduction*. MIT press Cambridge, 1998, vol. 1.

[11] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 336–361, First 2013.

[12] "5G: A technology vision," Huawei Technologies Co., White Paper, 2013.

[13] "5G vision," Samsung Electronics Co., White Paper, 2015.

[14] "Looking ahead to 5G," Nokia networks, White Paper, 2014.

[15] G. P. Fettweis, "A 5G wireless communications vision," *Microwave Journal*, vol. 55, no. 12, pp. 24–36, 2012.

[16] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[17] P. Wainio and K. Seppänen, "Self-optimizing last-mile backhaul network for 5G small cells," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 232–239.

[18] T. Alsedairy, Y. Qi, A. Imran, M. A. Imran, and B. Evans, "Self organising cloud cells: a resource efficient network densification strategy," *Transactions on Emerging Telecommunications Technologies*, vol. 26, no. 8, pp. 1096–1107, 2015. [Online]. Available: http://dx.doi.org/10.1002/ett.2824

[19] N. Baldo, L. Giupponi, and J. Mangues-Bafalluy, "Big data empowered self organized networks," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, May 2014, pp. 1–8.

[20] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, Nov 2014.

[21] "3rd generation partnership project; technical specification group services and system aspects; telecommunications management; self-organizing networks (SON); self-healing concepts and requirements (release 11), 3GPP TS 32.541, 2012-09, v11.0.0, 2012."

[22] "3GPP TS 36.902 evolved universal terrestrial radio access network (E-UTRAN); self-configuring and self-optimizing network (SON) use cases and solutions."

[23] "3GPP TS 32.500 telecommunication management; self-organizing networks (SON); concepts and requirements."

[24] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 190–199, October 2015.

[25] M. Marwangi, N. Fisal, S. Yusof, R. A. Rashid, A. S. Ghafar, F. A. Saparudin, and N. Katiran, "Challenges and practical implementation of self-organizing networks in LTE/LTE-Advanced systems," in *Information Technology and Multimedia (ICIM), 2011 International Conference on.* IEEE, 2011, pp. 1–5.

[26] J. P. B. Nadas, O. Onireti, R. D. Souza, H. Alves, G. Brante, and M. A. Imran, "Performance analysis of hybrid arq for ultra-reliable low latency communications," *IEEE Sensors Journal*, pp. 1–1, 2019.

[27] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.

[28] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile networks and applications*, vol. 19, no. 2, pp. 171–209, 2014.

[29] X. Wu, X. Zhu, G. Wu, and W. Ding, "Data mining with big data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, Jan 2014.

[30] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data: the management revolution," *Harvard business review*, vol. 90, no. 10, pp. 60–68, 2012.

[31] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning.* Springer series in statistics Springer, Berlin, 2001, vol. 1.

[32] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," 2007.

[33] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[34] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[35] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[36] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[37] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, ai-based, or both?" *arXiv preprint arXiv:1902.02647*, 2019.

[38] O. Onireti, A. Mohamed, H. Pervaiz, and M. Imran, "Analytical approach to base station sleep mode power consumption and sleep depth," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Oct 2017, pp. 1–7.

[39] P. V. Klaine, J. P. Nadas, R. D. Souza, and M. A. Imran, "Distributed drone base station positioning for emergency cellular networks using reinforcement learning," *Cognitive Computation*, pp. 1–15, 2018.

[40] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things journal*, vol. 1, no. 1, pp. 22–32, 2014.

[41] D.-W. Lee, G.-T. Gil, and D.-H. Kim, "A cost-based adaptive handover hysteresis scheme to minimize the handover failure rate in 3GPP LTE system," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, no. 1, p. 1, 2010.

[42] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta, "Massive mimo for next generation wireless systems," *arXiv preprint arXiv:1304.6690*, 2013.

[43] L. Lu, G. Y. Li, A. L. Swindlehurst, A. Ashikhmin, and R. Zhang, "An overview of massive mimo: Benefits and challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 742–758, Oct 2014.

[44] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5g cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, 2013.

[45] X. Gao, L. Dai, Y. Sun, S. Han, and I. Chih-Lin, "Machine learning inspired energy-efficient hybrid precoding for mmwave massive mimo systems," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[46] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[47] P. Simon, *Too Big to Ignore: The Business Case for Big Data*. John Wiley & Sons, 2013, vol. 72.

[48] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[49] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[50] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992. [Online]. Available: https://doi.org/10.1007/BF00992698

[51] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[52] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning." in *AAAI*, vol. 16, 2016, pp. 2094–2100.

[53] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[54] N. Heess, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, A. Eslami, M. Riedmiller *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[55] M. Jaber, M. A. Imran, R. Tafazolli, and A. Tukmanov, "Energy-efficient SON-based user-centric backhaul scheme," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, March 2017, pp. 1–6.

[56] D. Chen, J. Schuler, P. Wainio, and J. Salmelin, "5G self-optimizing wireless mesh backhaul," in *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2015, pp. 23–24.

[57] N. Farzaneh and M. H. Y. Moghaddam, "Virtual topology reconfiguration of WDM optical networks using fuzzy logic control," in *Telecommunications, 2008. IST 2008. International Symposium on*, Aug 2008, pp. 504–509.

[58] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: exploiting content caching and delivery techniques for 5G systems," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 131–139, February 2014.

[59] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, "Big data-driven optimization for mobile networks toward 5G," *IEEE Network*, vol. 30, no. 1, pp. 44–51, January 2016.

[60] P. Blasco and D. Gündüz, "Learning-based optimization of cache content in a small cell base station," in *2014 IEEE International Conference on Communications (ICC)*. IEEE, 2014, pp. 1897–1903.

[61] E. Baştuğ, M. Bennis, and M. Debbah, "A transfer learning approach for cache-enabled wireless networks," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2015 13th International Symposium on*, May 2015, pp. 161–166.

[62] M. S. ElBamby, M. Bennis, W. Saad, and M. Latva-Aho, "Content-aware user clustering and caching in wireless small cell networks," in *2014 11th*

International Symposium on Wireless Communications Systems (ISWCS). IEEE, 2014, pp. 945–949.

[63] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, 2014.

[64] K. Hamidouche, W. Saad, and M. Debbah, "Many-to-many matching games for proactive social-caching in wireless small cell networks," in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), 2014 12th International Symposium on*. IEEE, 2014, pp. 569–574.

[65] C. J. Debono and J. K. Buhagiar, "Cellular network coverage optimization through the application of self-organizing neural networks," in *VTC-2005-Fall. 2005 IEEE 62nd Vehicular Technology Conference, 2005.*, vol. 4. IEEE, 2005, pp. 2158–2162.

[66] H. Claussen, L. T. W. Ho, and L. G. Samuel, "Self-optimization of coverage for femtocell deployments," in *Wireless Telecommunications Symposium, 2008. WTS 2008*, April 2008, pp. 278–285.

[67] D. Fagen, P. A. Vicharelli, and J. Weitzen, "Automated wireless coverage optimization with controlled overlap," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2395–2403, 2008.

[68] A. Engels, M. Reyer, X. Xu, R. Mathar, J. Zhang, and H. Zhuang, "Autonomous self-optimization of coverage and capacity in LTE cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1989–2004, Jun 2013.

[69] H. Peyvandi, A. Imran, M. A. Imran, and R. Tafazolli, "On pareto-koopmans efficiency for performance-driven optimisation in self-organising networks," in *Intelligent Signal Processing Conference 2013 (ISP 2013), IET*, Dec 2013, pp. 1–6.

[70] L. T. W. Ho, I. Ashraf, and H. Claussen, "Evolving femtocell coverage optimization algorithms using genetic programming," in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2009, pp. 2132–2136.

[71] L. S. Mohjazi, M. A. Al-Qutayri, H. R. Barada, K. F. Poon, and R. M. Shubair, "Self-optimization of pilot power in enterprise femtocells using

multi objective heuristic," *Journal of Computer Networks and Communications*, vol. 2012, 2012.

[72] R. Joyce and L. Zhang, "Self organising network techniques to maximise traffic offload onto a 3G/WCDMA small cell network using MDT UE measurement reports," in *2014 IEEE Global Communications Conference*, Dec 2014, pp. 2212–2217.

[73] A. Gerdenitsch, S. Jakl, Y. Y. Chong, and M. Toeltsch, "A rule-based algorithm for common pilot channel and antenna tilt optimization in UMTS FDD networks," *ETRI journal*, vol. 26, no. 5, pp. 437–442, 2004.

[74] M. N. U. Islam and A. Mitschele-Thiel, "Reinforcement learning strategies for self-organized coverage and capacity optimization," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, April 2012, pp. 2818–2823.

[75] R. Razavi, S. Klein, and H. Claussen, "Self-optimization of capacity and coverage in LTE networks using a fuzzy reinforcement learning approach," in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2010, pp. 1865–1870.

[76] ——, "A fuzzy reinforcement learning approach for self-optimization of coverage in LTE networks," *Bell Labs Technical Journal*, vol. 15, no. 3, pp. 153–175, Dec 2010.

[77] H. Eckhardt, S. Klein, and M. Gruber, "Vertical antenna tilt optimization for LTE base stations," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–5.

[78] J.-H. Yun and K. G. Shin, "CTRL: a self-organizing femtocell management architecture for co-channel deployment," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 61–72.

[79] I. Karla, "Distributed algorithm for self organizing LTE interference coordination," in *International Conference on Mobile Networks and Management*. Springer, 2009, pp. 119–128.

[80] M. Mehta, N. Rane, A. Karandikar, M. A. Imran, and B. G. Evans, "A self-organized resource allocation scheme for heterogeneous macro-femto networks," *Wireless Communications and Mobile Computing*,

vol. 16, no. 3, pp. 330–342, 2016. [Online]. Available: http://dx.doi.org/10.1002/wcm.2518

[81] X. Zhao and P. Chen, "Improving UE SINR and networks energy efficiency based on femtocell self-optimization capability," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2014 IEEE*. IEEE, 2014, pp. 155–160.

[82] M. Bennis and D. Niyato, "A Q-learning based approach to interference avoidance in self-organized femtocell networks," in *2010 IEEE Globecom Workshops*, Dec 2010, pp. 706–710.

[83] M. Dirani and Z. Altman, "A cooperative reinforcement learning approach for inter-cell interference coordination in OFDMA cellular networks," in *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on*, May 2010, pp. 170–176.

[84] O. G. Aliu, M. Mehta, M. A. Imran, A. Karandikar, and B. Evans, "A new cellular-automata-based fractional frequency reuse scheme," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 4, pp. 1535–1547, April 2015.

[85] A. Quintero, "A user pattern learning strategy for managing users' mobility in UMTS networks," *IEEE Transactions on Mobile Computing*, vol. 4, no. 6, pp. 552–566, Nov 2005.

[86] A. Quintero and O. Garcia, "A profile-based strategy for managing user mobility in third-generation mobile systems," *IEEE Communications Magazine*, vol. 42, no. 9, pp. 134–139, 2004.

[87] K. Majumdar and N. Das, "Mobile user tracking using a hybrid neural network," *Wireless Networks*, vol. 11, no. 3, pp. 275–284, 2005. [Online]. Available: http://dx.doi.org/10.1007/s11276-005-6611-x

[88] S. Akoush and A. Sameh, "The use of Bayesian learning of neural networks for mobile user position prediction," in *Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007)*, Oct 2007, pp. 441–446.

[89] S. Premchaisawatt and N. Ruangchaijatupon, "Enhancing indoor positioning based on partitioning cascade machine learning models," in *Electrical Engineering/Electronics, Computer, Telecommunications and Informa-*

tion Technology (ECTI-CON), 2014 11th International Conference on, May 2014, pp. 1–5.

[90] J. Capka and R. Boutaba, "Mobility prediction in wireless networks using neural networks," in *IFIP/IEEE International Conference on Management of Multimedia Networks and Services*. Springer, 2004, pp. 320–333.

[91] M. Stoyanova and P. Mahonen, "A next-move prediction algorithm for implementation of selective reservation concept in wireless networks," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2007, pp. 1–5.

[92] M. Vukovic, I. Lovrek, and D. Jevtic, "Predicting user movement for advanced location-aware services," in *Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference on*, Sept 2007, pp. 1–5.

[93] X. Chen, F. Mériaux, and S. Valentin, "Predicting a user's next cell with supervised learning based on channel states," in *2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2013, pp. 36–40.

[94] V. S. Feng and S. Y. Chang, "Determination of wireless networks parameters through parallel hierarchical support vector machines," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 3, pp. 505–512, March 2012.

[95] A. Mohamed, O. Onireti, S. A. Hoseinitabatabaei, M. Imran, A. Imran, and R. Tafazolli, "Mobility prediction for handover management in cellular networks with control/data separation," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3939–3944.

[96] H. Si, Y. Wang, J. Yuan, and X. Shan, "Mobility prediction in cellular network using hidden Markov model," in *2010 7th IEEE Consumer Communications and Networking Conference*, Jan 2010, pp. 1–5.

[97] P. Fazio, M. Tropea, and S. Marano, "A distributed hand-over management and pattern prediction algorithm for wireless networks with mobile hosts," in *2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, July 2013, pp. 294–298.

[98] B. Sas, K. Spaey, and C. Blondia, "A SON function for steering users in multi-layer LTE networks based on their mobility behaviour," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, May 2015, pp. 1–7.

[99] ——, "Classifying users based on their mobility behaviour in LTE networks," in *ICWMC 2014, The Tenth International Conference on Wireless and Mobile Communications*, 2014.

[100] C. Yu, Y. Liu, D. Yao, L. T. Yang, H. Jin, H. Chen, and Q. Ding, "Modeling user activity patterns for next-place prediction," *IEEE Systems Journal*, vol. PP, no. 99, pp. 1–12, 2015.

[101] A. Chakraborty, L. E. Ortiz, and S. R. Das, "Network-side positioning of cellular-band devices with minimal effort," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, April 2015, pp. 2767–2775.

[102] H. Farooq and A. Imran, "Spatiotemporal mobility prediction in proactive self-organizing cellular networks," *IEEE Communications Letters*, vol. 21, no. 2, pp. 370–373, Feb 2017.

[103] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, "Predictive and core-network efficient RRC signalling for active state handover in RANs with control/data separation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1423–1436, March 2017.

[104] T. Jansen, I. Balan, J. Turk, I. Moerman, and T. Kurner, "Handover parameter optimization in LTE self-organizing networks," in *Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd*, Sept 2010, pp. 1–5.

[105] M. Peng, D. Liang, Y. Wei, J. Li, and H.-H. Chen, "Self-configuration and self-optimization in LTE-Advanced heterogeneous networks," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 36–45, 2013.

[106] D. Soldani, G. Alford, F. Parodi, and M. Kylvaja, "An autonomic framework for self-optimizing next generation mobile networks," in *2007 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2007, pp. 1–6.

[107] R. Narasimhan and D. C. Cox, "A handoff algorithm for wireless systems using pattern recognition," in *Personal, Indoor and Mobile Radio Communications, 1998. The Ninth IEEE International Symposium on*, vol. 1, Sep 1998, pp. 335–339 vol.1.

[108] P. P. Bhattacharya, A. Sarkar, I. Sarkar, and S. Chatterjee, "An ANN based call handoff management scheme for mobile cellular network," *CoRR*, vol. abs/1401.2230, 2014. [Online]. Available: http://arxiv.org/abs/1401.2230

[109] M. Ekpenyong, J. Isabona, and E. Isong, "Handoffs decision optimization of mobile celular networks," in *2015 International Conference on Computational Science and Computational Intelligence (CSCI)*, Dec 2015, pp. 697–702.

[110] N. Sinclair, D. Harle, I. A. Glover, J. Irvine, and R. C. Atkinson, "An advanced SOM algorithm applied to handover management within LTE," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1883–1894, Jun 2013.

[111] M. Stoyanova and P. Mahonen, "Algorithmic approaches for vertical handoff in heterogeneous wireless environment," in *2007 IEEE Wireless Communications and Networking Conference*, March 2007, pp. 3780–3785.

[112] I. Joe and S. Hong, "A mobility-based prediction algorithm for vertical handover in hybrid wireless networks," in *Broadband Convergence Networks, 2007. BcN '07. 2nd IEEE/IFIP International Workshop on*, May 2007, pp. 1–5.

[113] G. Hui and P. Legg, "Soft metric assisted mobility robustness optimization in LTE networks," in *2012 International Symposium on Wireless Communication Systems (ISWCS)*, Aug 2012, pp. 1–5.

[114] A. Schröder, H. Lundqvist, and G. Nunzi, "Distributed self-optimization of handover for the long term evolution," in *International Workshop on Self-Organizing Systems*. Springer, 2008, pp. 281–286.

[115] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, "A handover optimization algorithm with mobility robustness for LTE systems," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2011, pp. 1647–1651.

[116] L. Ewe and H. Bakker, "Base station distributed handover optimization in LTE self-organizing networks," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2011, pp. 243–247.

[117] I. Balan, T. Jansen, B. Sas, I. Moerman, and T. Kürner, "Enhanced weighted performance based handover optimization in LTE," in *Future Network Mobile Summit (FutureNetw), 2011*, June 2011, pp. 1–8.

[118] Q. Song, Z. Wen, X. Wang, L. Guo, and R. Yu, "Time-adaptive vertical handoff triggering methods for heterogeneous systems," in *International Workshop on Advanced Parallel Processing Technologies*. Springer, 2009, pp. 302–312.

[119] A. Awada, B. Wegmann, I. Viering, and A. Klein, "A SON-Based algorithm for the optimization of inter-RAT handover parameters," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1906–1923, Jun 2013.

[120] A. Beletchi, F. Huang, H. Zhuang, and J. Zha, "Mobility self-optimization in LTE networks based on adaptive control theory," in *2013 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2013, pp. 87–92.

[121] J. Alonso-Rubio, "Self-optimization for handover oscillation control in LTE," in *2010 IEEE Network Operations and Management Symposium - NOMS 2010*, April 2010, pp. 950–953.

[122] C. L. Lee, W. S. Su, K. A. Tang, and W. I. Chao, "Design of handover self-optimization using big data analytics," in *Network Operations and Management Symposium (APNOMS), 2014 16th Asia-Pacific*, Sept 2014, pp. 1–5.

[123] M. Toril and V. Wille, "Optimization of handover parameters for traffic sharing in GERAN," *Wireless personal communications*, vol. 47, no. 3, pp. 315–336, 2008.

[124] S. Luna-Ramirez, M. Toril, F. Ruiz, and M. Fernandez-Navarro, "Adjustment of a fuzzy logic controller for IS-HO parameters in a heterogeneous scenario," in *MELECON 2008 - The 14th IEEE Mediterranean Electrotechnical Conference*, May 2008, pp. 29–34.

[125] C. Werner, J. Voigt, S. Khattak, and G. Fettweis, "Handover parameter optimization in WCDMA using fuzzy controlling," in *2007 IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications*, Sept 2007, pp. 1–5.

[126] M. McGuire and V. K. Bhargava, "A robust fuzzy logic handoff algorithm," in *Electrical and Computer Engineering, 1997. Engineering Innovation:*

Voyage of Discovery. IEEE 1997 Canadian Conference on, vol. 2, May 1997, pp. 796–799 vol.2.

[127] L. Barolli, F. Xhafa, A. Durresi, A. Koyama, and M. Takizawa, "An intelligent handoff system for wireless cellular networks using fuzzy logic and random walk model," in *Complex, Intelligent and Software Intensive Systems, 2008. CISIS 2008. International Conference on*, March 2008, pp. 5–11.

[128] A. Ezzouhairi, A. Quintero, and S. Pierre, "A fuzzy decision making strategy for vertical handoffs," in *Electrical and Computer Engineering, 2008. CCECE 2008. Canadian Conference on.* IEEE, 2008, pp. 000 583–000 588.

[129] M. S. Dang, A. Prakash, D. K. Anvekar, D. Kapoor, and R. Shorey, "Fuzzy logic based handoff in wireless networks," in *Vehicular Technology Conference Proceedings, 2000. VTC 2000-Spring Tokyo. 2000 IEEE 51st*, vol. 3, 2000, pp. 2375–2379 vol.3.

[130] P. Munoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1895–1905, Jun 2013.

[131] S. S. Mwanje and A. Mitschele-Thiel, "Distributed cooperative Q-learning for mobility-sensitive handover optimization in LTE SON," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, vol. Workshops, June 2014, pp. 1–6.

[132] S. S. Mwanje, L. C. Schmelz, and A. Mitschele-Thiel, "Cognitive cellular networks: A Q-learning framework for self-organizing networks," *IEEE Transactions on Network and Service Management*, vol. 13, no. 1, pp. 85–98, March 2016.

[133] A. Quintero and S. Pierre, "On the design of large-scale UMTS mobile networks using hybrid genetic algorithms," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2498–2508, July 2008.

[134] V. Capdevielle, A. Feki, and A. Fakhreddine, "Self-optimization of handover parameters in LTE networks," in *Modeling Optimization in Mobile, Ad Hoc Wireless Networks (WiOpt), 2013 11th International Symposium on*, May 2013, pp. 133–139.

[135] F. Bouali, K. Moessner, and M. Fitch, "A context-aware user-driven framework for network selection in 5G multi-RAT environments," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Sept 2016, pp. 1–7.

[136] Z. Ali, N. Baldo, J. Mangues-Bafalluy, and L. Giupponi, "Machine learning based handover management for improved QoE in LTE," in *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, April 2016, pp. 794–798.

[137] C. Dhahri and T. Ohtsuki, "Cell selection for open-access femtocell networks: Learning in changing environment," *Physical Communication*, vol. 13, Part B, pp. 42 – 52, 2014, special Issue on Heterogeneous and Small Cell Networks. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1874490714000445

[138] ——, "Adaptive Q-learning cell selection method for open-access femtocell networks: Multi-user case," *IEICE Transactions on Communications*, vol. 97, no. 8, pp. 1679–1688, 2014.

[139] C. A. S. Franco and J. R. B. de Marca, "Load balancing in self-organized heterogeneous LTE networks: A statistical learning approach," in *2015 7th IEEE Latin-American Conference on Communications (LATINCOM)*. IEEE, 2015, pp. 1–5.

[140] I. Viering, M. Dottling, and A. Lobinger, "A mathematical perspective of self-optimizing wireless networks," in *2009 IEEE International Conference on Communications*, June 2009, pp. 1–6.

[141] A. Lobinger, S. Stefanski, T. Jansen, and I. Balan, "Load balancing in downlink LTE self-optimizing networks," in *Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st*, May 2010, pp. 1–5.

[142] J. Rodriguez, I. D. la Bandera, P. Munoz, and R. Barco, "Load balancing in a realistic urban scenario for LTE networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–5.

[143] P. Munoz, R. Barco, I. de la Bandera, M. Toril, and S. Luna-Ramirez, "Optimization of a fuzzy logic controller for handover-based load balancing," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd*, May 2011, pp. 1–5.

[144] P. Munoz, R. Barco, J. M. Ruiz-Avilés, I. de la Bandera, and A. Aguilar, "Fuzzy rule-based reinforcement learning for load balancing techniques in

enterprise LTE femtocells," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1962–1973, Jun 2013.

[145] S. S. Mwanje and A. Mitschele-Thiel, "A Q-learning strategy for LTE mobility load balancing," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2013, pp. 2154–2158.

[146] T. Kudo and T. Ohtsuki, "Q-learning based cell selection for UE outage reduction in heterogeneous networks," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–5.

[147] H. Hu, J. Zhang, X. Zheng, Y. Yang, and P. Wu, "Self-configuration and self-optimization for LTE networks," *IEEE Communications Magazine*, vol. 48, no. 2, pp. 94–100, 2010.

[148] H.-M. Zimmermann, A. Seitz, and R. Halfmann, "Dynamic cell clustering in cellular multi-hop networks," in *2006 10th IEEE Singapore International Conference on Communication Systems*. IEEE, 2006, pp. 1–5.

[149] M. Al-Rawi, "A dynamic approach for cell range expansion in interference coordinated LTE-advanced heterogeneous networks," in *Communication Systems (ICCS), 2012 IEEE International Conference on*. IEEE, 2012, pp. 533–537.

[150] L. Du, J. Bigham, L. Cuthbert, C. Parini, and P. Nahi, "Using dynamic sector antenna tilting control for load balancing in cellular mobile communications," in *International Conference on Telecommunications, ICT2002, Beijing*, vol. 2. Citeseer, 2002, pp. 344–348.

[151] S. Tomforde, A. Ostrovsky, and J. Hähner, "Load-aware reconfiguration of LTE-antennas dynamic cell-phone network adaptation using organic network control," in *Informatics in Control, Automation and Robotics (ICINCO), 2014 11th International Conference on*, vol. 01, Sept 2014, pp. 236–243.

[152] S. Bassoy, M. Jaber, M. A. Imran, and P. Xiao, "Load aware self-organising user-centric dynamic CoMP clustering for 5G networks," *IEEE Access*, vol. 4, pp. 2895–2906, 2016.

[153] P. Sandhir and K. Mitchell, "A neural network demand prediction scheme for resource allocation in cellular wireless systems," in *Region 5 Conference, 2008 IEEE*, April 2008, pp. 1–6.

[154] P. Fazio, F. D. Rango, and I. Selvaggi, "A novel passive bandwidth reservation algorithm based on neural networks path prediction in wireless environments," in *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*, July 2010, pp. 38–43.

[155] Y. Zang, F. Ni, Z. Feng, S. Cui, and Z. Ding, "Wavelet transform processing for cellular traffic prediction in machine learning networks," in *Signal and Information Processing (ChinaSIP), 2015 IEEE China Summit and International Conference on*, July 2015, pp. 458–462.

[156] I. Railean, C. Stolojescu, S. Moga, and P. Lenca, "Wimax traffic forecasting based on neural networks in wavelet domain," in *Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on*. IEEE, 2010, pp. 443–452.

[157] T. Edwards, D. Tansley, R. Frank, and N. Davey, *Traffic trends analysis using neural networks*, 1997, pp. 157–164.

[158] E. Ostlin, H. J. Zepernick, and H. Suzuki, "Macrocell radio wave propagation prediction using an artificial neural network," in *Vehicular Technology Conference, 2004. VTC2004-Fall. 2004 IEEE 60th*, vol. 1, Sept 2004, pp. 57–61 Vol. 1.

[159] A. Adeel, H. Larijani, A. Javed, and A. Ahmadinia, "Critical analysis of learning algorithms in random neural network based cognitive engine for LTE systems," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015, pp. 1–5.

[160] T. Binzer and F. M. Landstorfer, "Radio network planning with neural networks," in *Vehicular Technology Conference, 2000. IEEE-VTS Fall VTC 2000. 52nd*, vol. 2, 2000, pp. 811–817 vol.2.

[161] D. Kumar, N. Kanagaraj, and R. Srilakshmi, "Harmonized Q-learning for radio resource management in LTE based networks," in *ITU Kaleidoscope: Building Sustainable Communities (K-2013), 2013 Proceedings of*. IEEE, 2013, pp. 1–8.

[162] P. Savazzi and L. Favalli, "Dynamic cell sectorization using clustering algorithms," in *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring*. IEEE, 2007, pp. 604–608.

[163] A. Galindo-Serrano, L. Giupponi, and G. Auer, "Distributed learning in multiuser OFDMA femtocell networks," in *Vehicular Technology Conference (VTC Spring), 2011 IEEE 73rd.* IEEE, 2011, pp. 1–6.

[164] B. Fan, S. Leng, and K. Yang, "A dynamic bandwidth allocation algorithm in mobile networks with big data of users and networks," *IEEE Network*, vol. 30, no. 1, pp. 6–10, 2016.

[165] P. Kiran, M. Jibukumar, and C. Premkumar, "Resource allocation optimization in LTE-A/5G networks using big data analytics," in *2016 International Conference on Information Networking (ICOIN).* IEEE, 2016, pp. 254–259.

[166] A. Liakopoulos, A. Zafeiropoulos, C. Marinos, M. Grammatikou, N. Tcholtchev, and P. Gouvas, "Applying distributed monitoring techniques in autonomic networks," in *2010 IEEE Globecom Workshops.* IEEE, 2010, pp. 498–502.

[167] M. Dirani and Z. Altman, "Self-organizing networks in next generation radio access networks: Application to fractional power control," *Computer Networks*, vol. 55, no. 2, pp. 431–438, 2011.

[168] E. Alexandri, G. Martinez, and D. Zeghlache, "A distributed reinforcement learning approach to maximize resource utilization and control handover dropping in multimedia wireless networks," in *Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium on*, vol. 5. IEEE, 2002, pp. 2249–2253.

[169] D. Liu and Y. Zhang, "A self-learning adaptive critic approach for call admission control in wireless cellular networks," in *Communications, 2003. ICC '03. IEEE International Conference on*, vol. 3, May 2003, pp. 1853–1857 vol.3.

[170] L. T. Lee, C. F. Wu, D. F. Tao, and K. Y. Liu, "A cell-based call admission control policy with time series prediction and throttling mechanism for supporting QoS in wireless cellular networks," in *2006 International Symposium on Communications and Information Technologies*, Oct 2006, pp. 88–93.

[171] A. F. Santamaria and A. Lupia, "A new call admission control scheme based on pattern prediction for mobile wireless cellular networks," in *2015 Wireless Telecommunications Symposium (WTS)*, April 2015, pp. 1–6.

[172] T. Inaba, S. Sakamoto, T. Oda, and L. Barolli, "Performance evaluation of a secure call connection admission control for wireless cellular networks using fuzzy logic," in *2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA)*, Nov 2015, pp. 437–441.

[173] S. V. Truong, L. L. Hung, and H. N. Thanh, "A fuzzy logic call admission control scheme in multi-class traffic cellular mobile networks," in *2010 International Symposium on Computer, Communication, Control and Automation (3CA)*, vol. 1, May 2010, pp. 330–333.

[174] A. K. Mukhopadhyay, S. Chatterjee, S. Saha, S. Ghose, and D. Saha, "An efficient call admission control scheme on overlay networks using fuzzy logic," in *Advanced Networks and Telecommunication Systems (ANTS), 2009 IEEE 3rd International Symposium on*, Dec 2009, pp. 1–3.

[175] H. Nan, H. Zhiqiang, N. Kai, and W. Wei-ling, "Connection admission control for OFDM cellular networks by using fuzzy logic," in *2006 International Conference on Communication Technology*, Nov 2006, pp. 1–4.

[176] J. Ye, X. Shen, and J. W. Mark, "Call admission control in wideband CDMA cellular networks by using fuzzy logic," in *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, vol. 3, March 2003, pp. 1538–1543 vol.3.

[177] S. B. ZahirAzami, G. Yekrangian, and M. Spencer, "Load balancing and call admission control in UMTS-RNC, using fuzzy logic," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on*, vol. 2, April 2003, pp. 790–793 vol.2.

[178] P.-Y. Kong, D. Panaitopol, and A. Dhabi, "Reinforcement learning approach to dynamic activation of base station resources in wireless networks." in *PIMRC*, 2013, pp. 3264–3268.

[179] M. Peng and W. Wang, "An adaptive energy saving mechanism in the wireless packet access network," in *2008 IEEE Wireless Communications and Networking Conference*. IEEE, 2008, pp. 1536–1540.

[180] S. Samulevicius, T. B. Pedersen, and T. B. Sorensen, "MOST: mobile broadband network optimization using planned spatio-temporal events," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 2015, pp. 1–5.

[181] M. Miozzo, L. Giupponi, M. Rossi, and P. Dini, "Switch-on/off policies for energy harvesting small cells through distributed Q-learning," in *2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, March 2017, pp. 1–6.

[182] L. C. Wang, S. H. Cheng, and A. H. Tsai, "Bi-SON: Big-data self organizing network for energy efficient ultra-dense small cells," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Sept 2016, pp. 1–5.

[183] H. Y. Lateef, A. Imran, and A. Abu-dayya, "A framework for classification of self-organising network conflicts and coordination algorithms," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2013, pp. 2898–2903.

[184] A. Tall, R. Combes, Z. Altman, and E. Altman, "Distributed coordination of self-organizing mechanisms in communication networks," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 328–337, Dec 2014.

[185] H. Y. Lateef, A. Imran, M. A. Imran, L. Giupponi, and M. Dohler, "LTE-Advanced self-organizing network conflicts and coordination algorithms," *IEEE Wireless Communications*, vol. 22, no. 3, pp. 108–117, June 2015.

[186] I. Karla, "Resolving SON interactions via self-learning prediction in cellular wireless networks," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2012 8th International Conference on*, Sept 2012, pp. 1–6.

[187] S. Chia, M. Gasparroni, and P. Brick, "The next challenge for cellular networks: Backhaul," *IEEE Microw. Mag.*, vol. 10, no. 5, 2009.

[188] H. Galeana-Zapién and R. Ferrús, "Design and evaluation of a backhaul-aware base station assignment algorithm for OFDMA-based cellular networks," *IEEE Trans. Wireless Commun.*, vol. 9, no. 10, pp. 3226–3237, 2010.

[189] J. J. Olmos, R. Ferrus, and H. Galeana-Zapien, "Analytical modeling and performance evaluation of cell selection algorithms for mobile networks with backhaul capacity constraints," *IEEE Trans. Wireless Commun.*, vol. 12, no. 12, pp. 6011–6023, December 2013.

[190] C. Ran, S. Wang, and C. Wang, "Balancing backhaul load in heterogeneous cloud radio access networks," *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 42–48, June 2015.

[191] H. Elshaer, F. Boccardi, M. Dohler, and R. Irmer, "Load backhaul aware decoupled downlink/uplink access in 5G systems," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 5380–5385.

[192] A. D. Domenico, V. Savin, and D. Ktenas, "A backhaul-aware cell selection algorithm for heterogeneous cellular networks," in *2013 IEEE 24th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Sept 2013, pp. 1688–1693.

[193] F. Pantisano, M. Bennis, W. Saad, and M. Debbah, "Cache-aware user association in backhaul-constrained small cell networks," in *2014 12th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, May 2014, pp. 37–42.

[194] Q. Han, B. Yang, G. Miao, C. Chen, X. Wang, and X. Guan, "Backhaul-aware user association and resource allocation for energy-constrained hetnets," *IEEE Trans. Veh. Technol.*, vol. 66, no. 1, pp. 580–593, Jan 2017.

[195] M. Feng, S. Mao, and T. Jiang, "Joint frame design, resource allocation and user association for massive mimo heterogeneous networks with wireless backhaul," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 1937–1950, March 2018.

[196] H. Ma, H. Zhang, X. Wang, and J. Cheng, "Backhaul-aware user association and resource allocation for massive mimo-enabled hetnets," *IEEE Communications Letters*, vol. 21, no. 12, pp. 2710–2713, Dec 2017.

[197] Y. L. Lee, T. C. Chuah, A. A. El-Saleh, and J. Loo, "User association for backhaul load balancing with quality of service provisioning for heterogeneous networks," *IEEE Communications Letters*, pp. 1–1, 2018.

[198] G. Zhang, T. Q. Quek, M. Kountouris, A. Huang, and H. Shan, "Fundamentals of heterogeneous backhaul design—analysis and optimization," *IEEE Trans. Commun.*, vol. 64, no. 2, pp. 876–889, 2016.

[199] N. Bhushan, J. Li, D. Malladi, R. Gilmore, D. Brenner, A. Damnjanovic, R. Sukhavasi, C. Patel, and S. Geirhofer, "Network densification: the dominant theme for wireless evolution into 5G," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 82–89, 2014.

[200] P. Wang, W. Song, D. Niyato, and Y. Xiao, "QoS-aware cell association in 5G heterogeneous networks with massive MIMO," *IEEE Netw.*, vol. 29, no. 6, pp. 76–82, Nov 2015.

[201] Ericsson Review, "Heterogeneous network deployments in LTE - the soft-cell approach," *White Paper*, December 2011.

[202] T. S. Rappaport *et al.*, *Wireless communications: principles and practice.* prentice hall PTR New Jersey, 1996, vol. 2.

[203] S. Koenig and R. G. Simmons, "Complexity analysis of real-time reinforcement learning applied to finding shortest paths in deterministic domains," Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, Tech. Rep., 1992.

[204] "R1-103181: On range extension in open-access heterogeneous networks," *3GPP TSG RAN WG1 Meeting*, vol. 61, (Montreal, Canada), May 2010.

[205] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan 2017.

[206] J. Wang, Y. Wu, N. Yen, S. Guo, and Z. Cheng, "Big data analytics for emergency communication networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1758–1778, thirdquarter 2016.

[207] E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu, "On the number and 3D placement of drone base stations in wireless cellular networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, Sept 2016, pp. 1–6.

[208] H. Ahmadi, K. Katzis, and M. Z. Shakir, "A novel airborne self-organising architecture for 5g+ networks," in *Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th.* IEEE, 2017, pp. 1–5.

[209] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs," *IEEE Trans. Wireless Commun.*, vol. 15, no. 6, pp. 3949–3963, June 2016.

[210] I. Bor-Yaliniz and H. Yanikomeroglu, "The new frontier in RAN heterogeneity: Multi-tier drone-cells," *IEEE Commun. Mag.*, vol. 54, no. 11, pp. 48–55, November 2016.

[211] 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 Commun. Mag.*, vol. 54, no. 5, pp. 26–34, May 2016.

[212] A. Al-Hourani, S. Kandeepan, and S. Lardner, "Optimal LAP altitude for maximum coverage," *IEEE Wireless Commun. Lett.*, vol. 3, no. 6, pp. 569–572, Dec 2014.

[213] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone small cells in the clouds: Design, deployment and performance analysis," in *2015 IEEE Global Communications Conference (GLOBECOM)*, Dec 2015, pp. 1–6.

[214] 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 Commun. Lett.*, vol. 6, no. 4, pp. 434–437, Aug 2017.

[215] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," in *Wireless Communications and Networking Conference Workshops (WCNCW), 2015 IEEE*. IEEE, 2015, pp. 329–334.

[216] E. Kalantari, M. Z. Shakir, H. Yanikomeroglu, and A. Yongacoglu, "Backhaul-aware robust 3D drone placement in 5G+ wireless networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 109–114.

[217] M. M. Azari, F. Rosas, K. C. Chen, and S. Pollin, "Ultra reliable uav communication using altitude and cooperation diversity," *IEEE Trans. Commun.*, vol. 66, no. 1, pp. 330–344, Jan 2018.

[218] S. A. W. Shah, T. Khattab, M. Z. Shakir, and M. O. Hasna, "A distributed approach for networked flying platform association with small cells in 5g+ networks," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Dec 2017, pp. 1–7.

[219] J. Kennedy, "Particle swarm optimization," in *Encyclopedia of machine learning*. Springer, 2011, pp. 760–766.

[220] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, 2007.

[221] ITU-R, *Rec. P. 1410-2 Propagation Data and Prediction Methods for The Design of Terrestrial Broadband Millimetric Radio Access Systems*, 2003.

[222] ETSI Universal Mobile Telecommunication System (UMTS), "Selection procedures for the choice of radio transmission technologies of the UMTS," 1998-04, uMTS 30.03 Version 3.2.0. [Online]. Available: http://www.3gpp.org/ftp/Specs/html-info/3003U.htm

[223] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, "Modeling air-to-ground path loss for low altitude platforms in urban environments," in *2014 IEEE Global Communications Conference*, Dec 2014, pp. 2898–2904.

[224] Cisco Systems, "Cisco network emergency response vehicle," 2016, cisco Tactical Operations (TacOps). [Online]. Available: https://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/NERV_AAG.pdf

[225] M. M. Azari, F. Rosas, and S. Pollin, "Reshaping cellular networks for the sky: The major factors and feasibility," *arXiv preprint arXiv:1710.11404*, 2017.

[226] "Fullband fbxpmimo outdoor 4g antenna – cross polarised 3g/4g antenna," http://www.fullband.co.uk/product/fbxpmimo/, 2017, accessed: 2017-11-25.

[227] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[228] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *arXiv preprint arXiv:1803.00680*, 2018.

[229] Q. Zhang, M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Machine learning for predictive on-demand deployment of uavs for wireless communications," *arXiv preprint arXiv:1805.00061*, 2018.

[230] P. Yang, X. Cao, C. Yin, Z. Xiao, X. Xi, and D. Wu, "Proactive drone-cell deployment: Overload relief for a cellular network under flash crowd traffic," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 10, pp. 2877–2892, 2017.

[231] S. Sekander, H. Tabassum, and E. Hossain, "Multi-tier drone architecture for 5g/b5g cellular networks: Challenges, trends, and prospects," *IEEE Commun. Mag.*, vol. 56, no. 3, pp. 96–103, 2018.

[232] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, "Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046–1061, May 2017.

[233] V. Sharma, M. Bennis, and R. Kumar, "Uav-assisted heterogeneous networks for capacity enhancement," *IEEE Commun. Lett.*, vol. 20, no. 6, pp. 1207–1210, 2016.

[234] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Wireless communication using unmanned aerial vehicles (uavs): Optimal transport theory for hover time optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 8052–8066, 2017.

[235] M. Mozaffari, A. T. Z. Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5g with uavs: Foundations of a 3d wireless cellular network," *arXiv preprint arXiv:1805.06532*, 2018.

[236] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Drone-based antenna array for service time minimization in wireless networks," in *submitted to IEEE International Conference on Communications (ICC)*, 2018.

[237] F. Lagum, I. Bor-Yaliniz, and H. Yanikomeroglu, "Strategic densification with uav-bss in cellular networks," *IEEE Wireless Commun. Lett.*, vol. 7, no. 3, pp. 384–387, 2018.

[238] S. Koulali, E. Sabir, T. Taleb, and M. Azizi, "A green strategic activity scheduling for uav networks: A sub-modular game perspective," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 58–64, 2016.

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

[240] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-d placement of an unmanned aerial vehicle base station for maximum coverage of users with different qos requirements," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 38–41, 2018.

[241] K. Anazawa, P. Li, T. Miyazaki, and S. Guo, "Trajectory and data planning for mobile relay to enable efficient internet access after disasters," in *2015*

IEEE Global Communications Conference (GLOBECOM), Dec 2015, pp. 1–6.

[242] L. Wang, B. Hu, and S. Chen, "Energy efficient placement of a drone base station for minimum required transmit power," *IEEE Wireless Commun. Lett.*, pp. 1–1, 2018.

[243] M. Alzenad, M. Z. Shakir, H. Yanikomeroglu, and M. S. Alouini, "FSO-based vertical backhaul/fronthaul framework for 5G+ wireless networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 218–224, Jan 2018.

[244] R. Ghanavi, E. Kalantari, M. Sabbaghian, H. Yanikomeroglu, and A. Yongacoglu, "Efficient 3D aerial base station placement considering users mobility by reinforcement learning," in *Wireless Communications and Networking Conference (WCNC), 2018 IEEE*. IEEE, 2018, pp. 1–6.

[245] U. Challita, A. Ferdowsi, M. Chen, and W. Saad, "Artificial intelligence for wireless connectivity and security of cellular-connected UAVs," *arXiv preprint arXiv:1804.05348*, 2018.

[246] U. Challita, W. Saad, and C. Bettstetter, "Deep reinforcement learning for interference-aware path planning of cellular-connected UAVs," in *Proc. of International Conference on Communications (ICC). Kansas City, MO, USA*, 2018.

[247] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.

[248] M. Wiering and M. Van Otterlo, "Reinforcement learning," *Adaptation, learning, and optimization*, vol. 12, p. 3, 2012.

[249] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, J. Pineau *et al.*, "An introduction to deep reinforcement learning," *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.

[250] J. Heinrich and D. Silver, "Deep reinforcement learning from self-play in imperfect-information games," *arXiv preprint arXiv:1603.01121*, 2016.

[251] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, 2017.

[252] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in ofdm systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, 2018.

[253] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2019.

[254] T. Erpek, Y. E. Sagduyu, and Y. Shi, "Deep learning for launching and mitigating wireless jamming attacks," *IEEE Transactions on Cognitive Communications and Networking*, 2018.

[255] H. He, C.-K. Wen, S. Jin, and G. Y. Li, "Deep learning-based channel estimation for beamspace mmwave massive mimo systems," *IEEE Wireless Communications Letters*, vol. 7, no. 5, pp. 852–855, 2018.

[256] U. Challita, L. Dong, and W. Saad, "Proactive resource management for lte in unlicensed spectrum: A deep learning perspective," *IEEE Transactions on Wireless Communications*, vol. 17, no. 7, pp. 4674–4689, 2018.

[257] Y. Sun, M. Peng, and S. Mao, "Deep reinforcement learning based mode selection and resource management for green fog radio access networks," *IEEE Internet of Things Journal*, 2018.

[258] O. Onireti, A. Zoha, J. Moysen, A. Imran, L. Giupponi, M. A. Imran, and A. Abu-Dayya, "A cell outage management framework for dense heterogeneous networks," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2097–2113, April 2016.

[259] S. Chernov, D. Petrov, and T. Ristaniemi, "Location accuracy impact on cell outage detection in LTE-A networks," in *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Aug 2015, pp. 1162–1167.

[260] I. de-la Bandera, R. Barco, P. Munoz, and I. Serrano, "Cell outage detection based on handover statistics," *IEEE Communications Letters*, vol. 19, no. 7, pp. 1189–1192, July 2015.

[261] P. Muñoz, R. Barco, I. Serrano, and A. Gómez-Andrades, "Correlation-based time-series analysis for cell degradation detection in SON," *IEEE Communications Letters*, vol. 20, no. 2, pp. 396–399, 2016.

[262] H. Ye, Y. G. Li, and B.-H. F. Juang, "Deep reinforcement learning for resource allocation in v2v communications," *IEEE Transactions on Vehicular Technology*, 2019.

[263] H. Mao, M. Alizadeh, I. Menache, and S. Kandula, "Resource management with deep reinforcement learning," in *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*. ACM, 2016, pp. 50–56.

[264] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 11, pp. 10 433–10 445, 2017.

[265] L. T. Tan and R. Q. Hu, "Mobility-aware edge caching and computing in vehicle networks: A deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 190–10 203, 2018.