Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Telecommunication Engineering

Association of Users and Positioning of Flying Base Stations in Mobile Networks

Diploma thesis

Bc. Lukáš Vávra

Study programme: Electronics and Communications
Field of study: Communication Networks and Internet
Supervisor: prof. Ing. Zdeněk Bečvář, Ph.D.

Prague, January 2024

Thesis Supervisor:

prof. Ing. Zdeněk Bečvář, Ph.D.
Department of Telecommunication Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague
Technická 2
160 00 Prague 6
Czech Republic

ČVUT
ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE

ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Vávra** Jméno: **Lukáš** Osobní číslo: **474257**

Fakulta/ústav: **Fakulta elektrotechnická**

Zadávající katedra/ústav: **Katedra telekomunikační techniky**

Studijní program: **Elektronika a komunikace**

Specializace: **Komunikační sítě a internet**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Asociace uživatelů a výběr polohy létajících základnových stanic v mobilních sítích

Název diplomové práce anglicky:

Association of Users and Positioning of Flying Base Stations in Mobile Networks

Pokyny pro vypracování:

Seznamte se s konceptem létajících základnových stanic, tj. základnových stanic umístěných na bezpilotním letounu (UAV), pro budoucí mobilní sítě. Navrhněte způsob úpravy polohy létajících základnových stanic a asociace uživatelů k těmto stanicím v reálném čase tak, aby byla zajištěna komunikace pohybujících se uživatelů. Vzhledem ke složitosti problému a k požadavku na rozhodování v reálném čase, zvažte možnost využití strojového učení pro řešení obou problémů. Navržené řešení ověřte simulacemi a porovnejte jeho efektivitu s existujícími řešeními.

Seznam doporučené literatury:

[1] T. Sap, “Deployment of Flying Base Stations in Emergency Situations,” Diploma thesis, CTU in Prague, 2022.
[2] M. Najla, Z. Becvar, P. Mach and D. Gesbert, “Positioning and Association Rules for Transparent Flying Relay Stations,” IEEE Wireless Communications Letters, vol. 10, no. 6, 2021.
[3] M. Najla, Z. Becvar, P. Mach and D. Gesbert, “Predicting Device-to-Device Channels from Cellular Channel Measurements: A Learning Approach,” IEEE Transactions on Wireless Communications, vol. 19, no. 11, 2020.
[4] A. Fouda, A. S. Ibrahim, I. Guvenc, and M. Ghosh, “Interference Management in UAV-Assisted Integrated Access and Backhaul Cellular Networks,” IEEE Access, Vol. 7, 2019.

Jméno a pracoviště vedoucí(ho) diplomové práce:

doc. Ing. Zdeněk Bečvář, Ph.D. katedra telekomunikační techniky FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) diplomové práce:

Datum zadání diplomové práce: **18.09.2023** Termín odevzdání diplomové práce: **09.01.2024**

Platnost zadání diplomové práce: **16.02.2025**

doc. Ing. Zdeněk Bečvář, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Diplomant bere na vědomí, že je povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v diplomové práci.

.
Datum převzetí zadání

Podpis studenta

Declaration

I hereby declare I have written this diploma thesis independently and quoted all the sources of information used in accordance with methodological instructions on ethical principles for writing an academic thesis. Moreover, I state that this thesis has neither been submitted nor accepted for any other degree.

In Prague, January 2024

....
Bc. Lukáš Vávra

Abstract

Mobile networks in which unmanned aerial vehicles (UAVs) are deployed and serve as flying base stations (FlyBSs) can potentially increase channel capacities offered to mobile network users and therefore improve the overall mobile network performance. The optimization of the performance of the mobile networks with the integrated FlyBSs from the perspective of the FlyBS positioning, the user equipment (UE) association to the FlyBSs or the transmission power allocation etc. has already been investigated in multiple research works. This thesis, however, proposes a novel framework that jointly addresses the problem of the positioning of the FlyBSs and the problem of the UE association via introducing offline-trained deep neural networks (DNNs). The proposed framework integrates the DNNs into a system of interconnected DNNs that cooperate with each other by sharing prediction information in order to improve the accuracies of their predictions and therefore to enhance the overall performance of the proposed framework. The simulation results show that the proposed framework outperforms the other competitive schemes in terms of the overall channel capacity by 14% - 95% depending on the number of the FlyBSs and the number of the UEs in the mobile network cell. In terms of fairness of the distribution of the channel capacities among the UEs in the mobile network measured by Jain's fairness index, the proposed framework delivers better results by 5% - 106% in comparison with the competitive schemes.

Keywords: mobile network, flying base station, deep neural network, positioning of flying base stations, association of user equipment

Abstrakt

Mobilní sítě, které využívají nasazení bezpilotních vzdušných letounů (UAVs), jež slouží jako létající základnové stanice (FlyBSs), mohou potenciálně zvýšit kapacity kanálů, které jsou poskytované uživatelům, a tím i celkově zlepšit výkonnost mobilní sítě. Metody zlepšení výkonnosti mobilních sítí s integrovanými FlyBSs z hlediska rozmístění FlyBSs v buňce mobilní sítě, asociování uživatelských zařízení (UEs) k FlyBSs nebo alokace vysílacího výkonu atd. již byly analyzovány v mnoha výzkumných pracích. Tato diplomová práce ovšem prezentuje nový návrh řešení optimalizace výkonnosti mobilní sítě, který společně řeší problém rozmístění FlyBSs zároveň s problémem asociace UEs, a to prostřednictvím hlubokých neuronových sítí (DNNs). Návrh integruje DNNs do komplexního systému propojených DNNs, které mezi sebou sdílí informace s cílem zpřesnit své predikce, a tím i zvýšit celkovou výkonnost návrhu. Výsledky simulací ukazují, že návrh překonává konkurenční schémata z hlediska celkové kapacity kanálu o 14% - 95% v závislosti na počtu FlyBSs a počtu UEs v buňce mobilní sítě. Pokud jde o férovost distribuce kapacit kanálů mezi UEs v mobilní síti měřenou Jainovým indexem férovosti, navržené řešení prezentuje lepší výsledky ve srovnání s konkurenčními schématy, a to konkrétně o 5% - 106% opět v závislosti na na počtu FlyBSs a UEs v buňce mobilní sítě.

Klíčová slova: mobilní síť, létající základnová stanice, hluboká neuronová síť, rozmístění létajících základnových stanic, asociace uživatelských zařízení

Acknowledgements

I would like to express my gratitude to prof. Ing. Zdeněk Bečvář, PhD. for his supervision, for precious advice and ideas which showed me the right direction many times. Furthermore, I would also like to thank my family and especially Kristýna Šeredová for their unconditional support during my studies and writing this thesis.

List of Tables

List of Figures

List of Abbreviations

3GPP	3rd Generation Partnership Project
BS	Base Station
DNN	Deep Neural Network
FlyBS	Flying Base Station
GBS	Ground Base Station
IAB	Integrated Access and Backhaul
LOS	Line-Of-Sight
NLOS	Non-Line-Of-Sight
ReLU	Rectified Linear Unit
UAV	Unmanned Aerial Vehicle
UE	User Equipment

Contents

Chapter 1

Introduction

Since the rise of mobile networking, constantly increasing requirements on throughput, coverage, low latency, or reliability have been observed. These requirements and demands have played a key role in developing new technologies, new network architectures, and generally new concepts that could lead to an improvement in the performance of the mobile network. One of the concepts that aim to satisfy emerging demands is to deploy unmanned aerial vehicles (UAV) [1]. Utilizing the UAVs with integrated base stations, in this thesis referred as flying base stations (FlyBS), in the future mobile networks complementing ground base stations (GBS) aims to be a suitable and flexible concept that could assist in fulfilling increasing demands of mobile network users [2], [3]. The FlyBSs connected to the GBSs via relay links are able to extend the coverage of the mobile network or improve the quality of provided services in a specific geographical area by adapting their positions according to network's needs [4], [5]. One of the main benefits of the FlyBSs then is that their deployment can be fast which results in the network being able to react to quickly changing situations, such as short-time traffic peaks [4], [6].

These qualities indicate that deploying the FlyBSs in the mobile networks can be potentially beneficial for providing broadband and wide-area temporary wireless connectivity in situations where the permanent instalment of ground networking infrastructures is unjustified in terms of financial or other reasons [7]. These situations cover scenarios in which there is a high density of the mobile network users in a certain location, such as traffic jams, big sport events or concerts [4], [7]. The traffic load originated due to the mentioned events is temporary and after the events, the load is spread into a larger area. The concept of deploying the FlyBSs assumes that the FlyBSs are able to temporarily serve user equipment (UE) in a given location with an increased traffic load. When the load decreases, the FlyBSs can be relocated to another position where needed.

The challenges that are also addressed by the deployment of the FlyBSs are, for instance, serving remote locations with insufficient coverage due to various economical or geographical reasons, such as smart farming [7]. Other use cases may include emergency situations that could be caused by, for instance, natural disasters, such as floods, earthquakes, or tornados etc., which could cause damage on the network infrastructure and therefore cause an outage of reliable connection [7].

Although the potential of the concept of the FlyBS in the mobile networks is significant, it raises several problems that need to be addressed before an introduction of the FlyBSs to the mobile networks. Among these problems, a positioning of the FlyBSs in a cell area, a distribution of the UEs among the FlyBSs, an optimization of trajectories of movements of the FlyBSs, a transmission power allocation between the BS and the FlyBSs and between the FlyBS and the associated UEs, a bandwidth allocation or a reasonable handover policy should be mentioned [4], [6].

Finding an optimized positioning of the FlyBSs in the mobile networks altogether with determining an appropriate UE association strategy belongs among the key challenges to be addressed. Optimization of both processes leads to a significant increase in the quality and the reliability of the provided services [7], [8]. Some works, such as [3], [9] or [10], combine both the FlyBS positioning optimization as well as the UE association optimization into a general framework whose objective is to improve the performance of the mobile networks. While [9] and [10] propose a promising solution to tackle the above-mentioned challenges considering only connection parameters related to access links between UEs and FlyBSs, the authors in [3] also consider the backhaul links. The concept of the integrated access and backhaul (IAB) network architecture is introduced by 3rd Generation Partnership Project (3GPP) in [11]. The IAB network architecture concept assumes a close cooperation between the access link and the backhaul link, i.e. the link between the FlyBSs and the BS. The IAB network architectures raise a challenge of the mutual interference between the access and the backhaul links. This challenge is targeted in [3], where the authors propose an interference management algorithm whose objective is to optimize the UE association and the power allocation of both the access links and the backhaul links. The algorithm in [3] provides high capacities to the UEs and is power-efficient, but the allocated power for the backhaul links and for the UEs directly associated with the BS does not differ which leads to limited efficiency of the FlyBSs in cases when the majority of the UEs are associated directly with the BS.

The work [12] also builds on the IAB network principles and presents an algorithm for the UE association based on the K-means clustering followed by reassociation steps in order to optimize the distribution of the UEs among the FlyBSs and the GBS and to eliminate bottlenecks caused by a possible overload of certain FlyBSs. The positioning problem in [12] is addressed via an algorithm based on an adaptation of Coulomb's law, where virtual attracting forces originating in the UEs and in the BS act on the FlyBSs.

This concept with an objective of maximizing efficiency of the FlyBSs deployment in terms of satisfying the UE data rate requirements while considering the energy consumption of the deployed FlyBSs presented in [12] is, however, mainly focused on maximizing the minimal channel capacity among the UEs and therefore does not work as a suitable solution for maximizing the overall capacity in the mobile network.

The frameworks presented in [5], [9] and [13] and [14] assume that exact locations of the UEs in the mobile network cell are known. However, as mentioned in [6], in practice, the locations of the UEs are usually inaccurate or remain undisclosed to the network due to privacy requirements. This raises another challenge that needs to be addressed. The paper [2] investigates deploying the FlyBSs as so-called transparent relays in the mobile networks in cases where the UE locations are known to the network, but also for cases where the UE locations remain undisclosed. The authors in [2] explore possibilities of utilizing deep neural networks (DNN) to predict channel gains between the UEs and the FlyBSs that are then used for the optimization of the UE associations. However, the number of UEs whose associations are predicted by the DNN goes only up to 5. The number of UEs in the mobile network cell is generally much greater. A similar deep-learning approach applied to a greater number of the UEs in the mobile network is investigated in [6] where the DNNs are introduced to solve the positioning of the FlyBSs and the association of the UEs in urban scenarios with the UE locations being unknown.

Utilizing the DNNs for estimating the channel gains is also investigated in [15] where it is shown that the DNNs are capable of estimating the channel gains not only for line-of-sight (LOS) scenarios, but also for non-line-of-sight (NLOS) scenarios. Although the applicability of the DNNs to predict channel qualities is proven in [15], this work focuses on investigation of the DNN utilization in the device-to-device communications which differs from the mobile network communications investigated in this thesis.

The main objective of this thesis is to increase the overall channel capacity offered to the UEs in the cell of the mobile cellular network in the LOS scenarios. For this purpose, this thesis presents a solution that optimizes the processes of the positioning of FlyBSs and the association of UEs via introducing principles of supervised deep learning methods and integrates both processes into a robust framework. The framework combines three complex offline-trained DNNs, optimizing the positioning and the association respectively in order to fulfill the objectives of the thesis. The simulation results show that the proposed framework outperforms the other competitive schemes by 14% - 95% in terms of the overall channel capacity depending on the number of the FlyBSs and the number of the UEs in the mobile network cell. In terms of fairness of the distribution of the channel capacities among the UEs measured by Jain's fairness index, the proposed framework reports better results by up to 5% - 106% in comparison with the competitive schemes.

In order to comprehensively present the proposed solution, this diploma thesis is organized as follows. Chapter 2 presents the description of the system model covering the model of the network, the channel model, the concept of virual shift forces applied to the FlyBSs, the modified K-means clustering and high-level DNN principles. Chapter 3 states the thesis problem formulation. Chapter 4 describes the algorithms that address the FlyBS positioning and the UE association subproblems. In Chapter 4, these algorithms are subsequently combined into the robust final framework that comprehensively addresses the problem formulated in Chapter 3. In Chapter 5, the parameters of the system model are presented altogether with other competitive schemes and with metrics that are later used for the framework performance evaluation. Chapter 6 presents the results of the performance evaluation based on simulations of the proposed framework and the competitive schemes. Finally, Chapter 7 summarizes the results of the performance evaluation and presents possible directions of future research.

Chapter 2

System Model

In this chapter, the whole system model utilized in this thesis is described. At the beginning of this chapter, the network model is presented. This is followed by the description of the channel model where the channel parameters between the objects located in the network model are defined. The network model and the channel model are based on the work of T. Sap presented in [12]. Subsequently, the concept of the virtual shift forces that act on the FlyBSs and the algorithm of the modified K-means clustering are clarified. Finally, the principles of functioning of the DNNs are explained. As the topic of the DNNs is very broad, the attention is mainly focused on the basic principles and parameters that are essential for understanding of the proposed final solution.

2.1 Network model

In this section, two parts of the network model are described. First, the system space, with the objects deployed in the network model, is defined. This is followed by the definition of the links that are established between the system objects and the definition of the UE associations.

2.1.1 System Space

The system space represents an area that is covered by the cellular mobile network. This means that the system space is divided into cells where each cell is governed by a single BS from a set of N_{BS} base stations $\mathbb{B} = \{\text{BS}_1, \text{BS}_2, \ldots, \text{BS}_{N_{BS}}\}$. In the network model, two types of the cells are defined: one reference cell and the other cells that neighbour the reference cell and introduce intercell interference. The reference cell is a two-dimensional space with its dimension lengths defined as d_1 and d_2 and includes objects of three types: the ground base station BS_1 (in this thesis referred as the BS or the serving BS), a set of N_{FlyBS} flying base stations $\mathbb{F} = \{\text{FlyBS}_1, \text{FlyBS}_2, \ldots, \text{FlyBS}_{N_{FlyBS}}\}$ and a set of N_{UE} user equipment $\mathbb{U} = \{\text{UE}_1,\text{UE}_2,\ldots,\text{UE}_{N_{UE}}\}$. The serving BS together with the set $\mathbb{F}$ and the set $\mathbb{U}$ forms a set that is in this thesis referred as the cell objects. The neighboring cells are represented by the remaining BSs from $\mathbb{B}$ and impose interference on the reference cell.

The positions of the above-mentioned objects are generally three-dimensional and are defined by the coordinates x, y and z. The positions of the BSs are then defined as $\mathbb{R}_{BS} = \{\mathbf{r_{BS_1}},\mathbf{r_{BS_2}},\ldots,\mathbf{r_{N_{BS}}}\}$ where $\mathbf{r_{BS_k}} = \{x_{BS_k},\, y_{BS_k},\, z_{BS_k}\}$ for BS_k. In a similar way as the positions of the BSs, the positions of the FlyBSs in the cell are defined as $\mathbb{R}_{FlyBS} = \{\mathbf{r_{FlyBS_1}},\mathbf{r_{FlyBS_2}},\ldots,\mathbf{r_{N_{FlyBS}}}\}$ where $\mathbf{r_{FlyBS_j}} = \{\ x_{FlyBS_j},\, y_{FlyBS_j},\, z_{FlyBS_j}\}$ for FlyBS_j. Analogically, the positions of UEs are defined as $\mathbb{R}_{UE} = \{\mathbf{r_{UE_1}},\mathbf{r_{UE_2}},\ldots,\mathbf{r_{UE_{N_{UE}}}}\}$ where $\mathbf{r_{UE_i}} = \{x_{UE_i},\, y_{UE_i},\, z_{UE_i}\}$ for UE_i.

The coordinate z expresses an altitude of the cell objects above the ground. Therefore, for the BSs, the values of their coordinates z remain constant and represent the height of the BSs h_{BS}. For the sake of simplicity, the values of coordinates z for each UE are also unified, and thus their value equals to h_{UE}. The z coordinates of the FlyBSs are not fixed, although their range is limited by the value of the minimum FlyBS altitude $z_{FlyBS_{min}}$ and the maximum FlyBS altitude $z_{FlyBS_{max}}$.

Regarding the coordinates x and y, for the purposes of this thesis, all UE are considered static, i.e., the coordinates x and y of the UEs are determined at the beginning of each simulation scenario and do not change throughout the simulation. On the other hand, the FlyBSs are allowed to change their coordinates, as it is one of their most obvious benefits [4]. The positions of the BSs also remain static. The location of the serving BS is fixed to the center of the cell.

2.1.2 Links between System Objects

In the cellular mobile network, the BS serves as the central connection point for the UEs to communicate with the other UEs or other network elements. Therefore, in order to be served by the network, the UE has to be connected to the BS.

In the proposed network model, there are two possible cases how the UE can be connected to the BS. In the first scenario, the UE is associated to the BS directly forming a direct link. In the second case, the UE is associated to the FlyBS that is then associated to the BS. The FlyBS forwards information from the BS to the UE and vice versa and acts as a relay. This setup forms a relay link. The relay link is composed of the access link, i.e. the link between the UE and the FlyBS, and the backhaul link, i.e. between the FlyBS and the BS [3].

2.1.3 UE Associations

The links that are established in the system model are described via sets of associations. Each UE holds two sets of associations, one with the BSs and the other with the Fly-BSs. For the $\mathrm{UE_i}$, the set of the BS associations is defined as $\mathbf{a_{UE_i,BS}} = \{a_{UE_i,BS_1}, \ldots, a_{UE_i,BS_{N_{BS}}}\}$ and the set of the FlyBS associaitons is defined as $\mathbf{a_{UE_i,FlyBS}} = \{ a_{UE_i,FlyBS_1}, \ldots, a_{UE_i,FlyBS_{N_{FlyBS}}}\}$. The sets $\mathbf{a_{UE_i,BS}}$ and $\mathbf{a_{UE_i,FlyBS}}$ are then megred into the set $\mathbf{a_{UE_i}}$ that unites all the possible UE associations. For the $\mathrm{UE_i}$, the values of the elements of $\mathbf{a_{UE_i}}$ are then defined as follows:

- $a_{UE_{i,n}} = 1$ if the UE_i is associated with the BS, or the FlyBS that is represented by the n-th element of the set $\mathbf{a_{UE_i}}$,
- $a_{UE_{i,n}} = 0$ otherwise.

Note that each UE has to be connected either directly to the BS or to one and only FlyBS. This is expressed by the following constraint:

$$\sum \mathbf{a_{UE_i}} = 1 \tag{2.1}$$

In total, an association scenario is then described by N_{UE} association sets. By merging the UE association sets, the association matrix $\mathbb{A}$ is created. The association matrix $\mathbb{A}$ is described as $\mathbb{A} = \{\mathbf{a_{UE_1}}, \ldots, \mathbf{a_{UE_{N_{UE}}}}\}$. Figure 2.1 shows the system model with the possible associations.

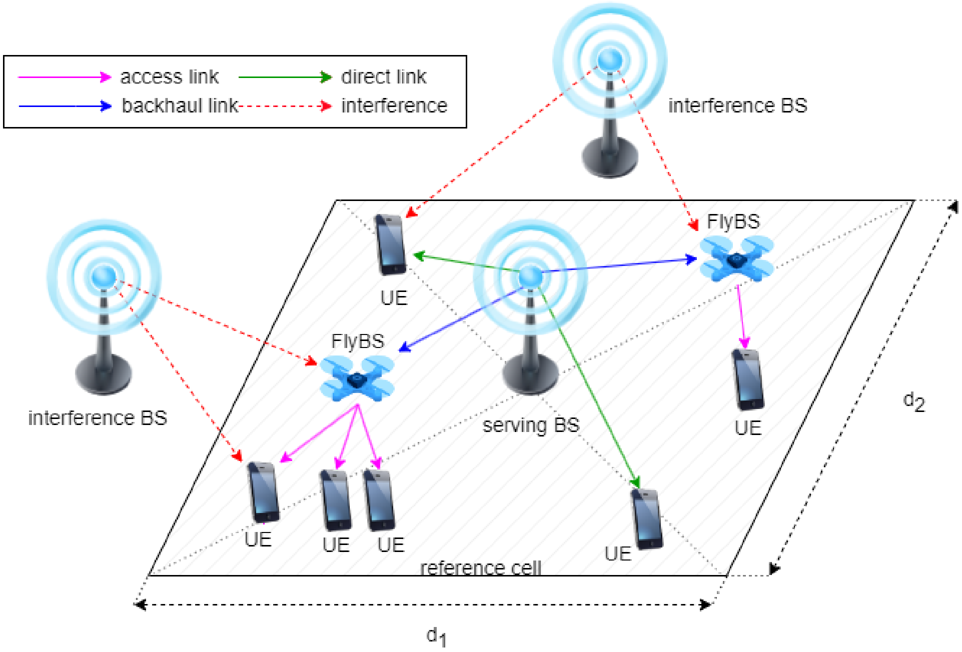


Figure 2.1: Example of the system model.

2.2 Channel Model

In this section, the characteristics of the communication channel are clarified. First, the parameter of the channel gain is defined. This is followed by the definition of the bandwidth allocation, the transmission power allocation, the interference and the communication channel capacity.

2.2.1 Channel Gain

In the system model, no obstacle is introduced to the reference cell. Therefore, the environment of the model is considered LoS. The channel gain expresses a difference in the signal level between a transmitting network element and an element that receives the signal. The channel gain is determined by the following formula:

$$g = (\frac{c}{4\pi df})^2 \tag{2.2}$$

where g is the channel gain, c stands for the speed of light, d represents the distance between the transmitter and the receiver and f is the frequency of the carrier.

2.2.2 Bandwidth Allocation

Each cell in the system model, i.e. the reference cell and all the neighbouring cells, operate in the same frequency band and also share the same bandwidth B_{total}. The bandwidth in the reference cell is then divided into equal smaller bandwidths and distributed among all the UEs. This can be expressed by the following formula:

$$B_{UE} = \frac{B_{total}}{N_{UE}} \tag{2.3}$$

where B_{UE} is the bandwidth allocated to one UE and N_{UE} stands for the number of the UEs in the reference cell.

In case of the relayed links, the access link and the backhaul link share the same communication channel with the bandwidth B_{UE} as the bandwidth allocation optimization is not the main objective of this thesis.

2.2.3 Transmission Power Allocation

In the system model, the BS and each FlyBS have dedicated power budgets that are distributed among their associated objects. The power budget of the serving BS is denoted as P_{BS}, whereas the power budget of all FlyBSs is unified and its value is denoted as P_{FlyBS}. The power allocation policy then adopts the rules in accordance with which type of link is considered. In case of the direct link, the allocated power from the BS towards the UE$_\mathrm{i}$ is defined as:

$$p^{(D)}_{UE_i} = \frac{P_{BS}}{N_{UE}} \tag{2.4}$$

where N_{UE} stands for the number of the UEs in the reference cell.

In case of the backhaul link, the power allocated from the BS towards the FlyBS is dependent on the number of UEs that are associated to the FlyBS. The power allocated to the FlyBS$_\mathrm{j}$ is calculated as:

$$p_{FlyBS_j} = \frac{P_{BS}}{N_{UE}} N_{UE_{FlyBS_j}} \tag{2.5}$$

where $N_{UE_{FlyBS_j}}$ represents the number of UEs associated to the FlyBS$_\mathrm{j}$.

Finally, for the access link, the power allocated to the UE$_\mathrm{i}$ from the associated FlyBS$_\mathrm{j}$ is evaluated as:

$$p^{(A)}_{UE_i} = \frac{P_{FlyBS}}{N_{UE_{FlyBS_j}}} \tag{2.6}$$

where P_{FlyBS} stands for the power budget of the FlyBSs and $N_{UE_{FlyBS_j}}$ represents the number of UEs associated to the FlyBS$_\mathrm{j}$. Figure 2.2 illustrates the transmission power allocation scheme.

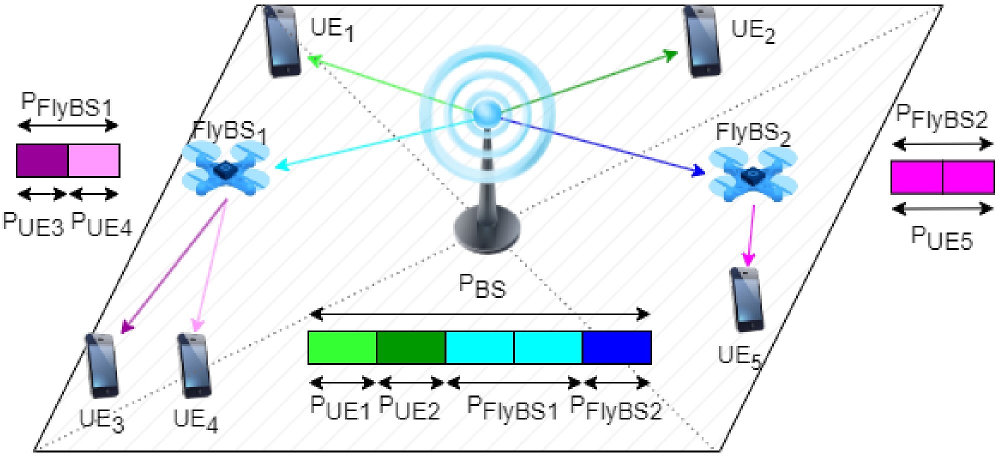


Figure 2.2: Example of the power allocation in the reference cell.

2.2.4 Interference

Each communication channel is affected by interference. In the system model, two types of interference are considered: intracell interference, caused by objects that are located within the reference cell, and intercell interference, originating in the objects that are located outside the reference cell borders in the neighbouring cells. In the proposed system model, the interference power allocated to all channels in the neighbouring cells is estimated as:

$$P_I = \frac{P_{BS}}{N_{UE}} \tag{2.7}$$

where P_{BS} is the power budget of the BS and N_{UE} is the number of the UEs in the reference cell.

In the system model, the UE can be connected via the direct link to the BS. In this case the signal-to-interference-and-noise ratio (SINR) is evaluated as follows:

$$SINR_D = \frac{p_i g_{b,i}}{\sigma B_{UE} + \sum_{v \in \mathbb{B}_I} P_I g_{m,i}} \tag{2.8}$$

where p_i is the downlink transmission power allocated towards the $\mathrm{UE_i}$, $g_{b,i}$ represents the channel gain between the BS and the $\mathrm{UE_i}$, σ stands for the noise spectral density, B_{UE} is the bandwidth allocated to the UE from the BS, $\mathbb{B}_I$ stands for the set of interfering BSs, P_I represents the transmission power of the interfering signal and $g_{m,i}$ is the channel gain between the $\mathrm{UE_i}$ and the neighbouring BS where P_I originates.

When the UE is connected via the relay link, both SINR for the access link and for the backhaul link are considered. For the backhaul link the SINR is evaluated as follows:

$$SINR_B = \frac{p_j g_{b,j}}{\sigma B_{UE} + \sum_{v \in \mathbb{B}_I} P_I g_{m,j}} \tag{2.9}$$

where p_j is the downlink transmission power allocated towards the $\mathrm{FlyBS_j}$ from the serving BS, $g_{b,j}$ represents the channel gain between the FlyBS and the BS and $g_{m,j}$ is the channel gain between the $\mathrm{FlyBS_j}$ and the neighbouring BS where P_I originates.

Finally, the access link SINR is expressed as:

$$SINR_A = \frac{p_i g_{j,i}}{\sigma B_{UE} + p_j g_{b,i} \sum_{v \in \mathbb{B}_I} P_I g_{m,i}} \tag{2.10}$$

where p_i is the downlink transmission power allocated towards the UE from the FlyBS, $g_{j,i}$ represents the channel gain between the UE and the FlyBS.

2.2.5 Channel Capacity

To evaluate the channel capacities, Shannon's theorem is utilized. The channel capacities for the direct (D), access (A) and backhaul (B) links are then calculated as follows:

$$C_D = B_{UE} log_2(1 + SINR_D) \tag{2.11}$$

$$C_A = B_{UE} log_2(1 + SINR_A) \tag{2.12}$$

$$C_B = B_{UE} log_2(1 + SINR_B) \tag{2.13}$$

where B_{UE} stands for the bandwidth allocated to the UE from the FlyBS and SINR represents the signal-to-noise-to-interference ratio of the corresponding direct (D), access (A) or backhaul (B) link.

Generally, the capacities of the access link and the backhaul link do not have the same value, thus the capacity of the relay link (R) is the lower value from those two values. Formally, this is expressed as:

$$C_R = min(C_A, C_B) \tag{2.14}$$

2.3 Virtual Shift Forces

The channel capacities for each UE are influenced by the distance from the serving station, either the FlyBS or the BS which the UE is associated to. Logically, when the distance between the UE and the BS is shorter, the channel capacity tends to improve. However, in case the UE is associated to the FlyBS, more factors must be considered. In this case, not only the access link between the UE and the FlyBS impacts the channel capacity, but also the backhaul link between the FlyBS and the associated UE must be taken into account. Low quality backhaul links might not offer a sufficient channel capacity between the FlyBS and the associated BS and therefore might create a bottleneck negatively impacting the overall relay channel quality.

The concept introduced in [5] addresses the problem of finding the optimized positions for the FlyBSs considering both the access link and the backhaul link qualities. This concept introduces a system of virtual forces, similar to electrostatic forces, that act on the FlyBSs. In order to determine the optimized positions of the FlyBSs, the work [5] defines two types of the virtual forces. The BS, which the FlyBS is associated to, interacts with the FlyBS and attracts the FlyBS with a backhaul force. The FlyBSs also experience access forces that originate in the associated UEs. However, the work [5] defines the virtual forces based on the power allocation parameters in the reference cell. Unlike that, this thesis presents a concept of the virtual force definition based on the distances between the FlyBS and the UEs and the BS. The backhaul force between the $\mathrm{FlyBS_j}$ and the BS is expressed by the following formula:

$$\overrightarrow{F_{j,b}^{(B)}} = \frac{\mathbf{r_{BS}} - \mathbf{r_{FlyBS_j}}}{\left\|\mathbf{r_{BS}} - \mathbf{r_{FlyBS_j}}\right\|} \tag{2.15}$$

where the symbol $\mathbf{r_{BS}}$ stands for the coordinates of the BS in the reference cell and $\mathbf{r_{FlyBS_j}}$ represent coordinates of the $\mathrm{FlyBS_j}$. The operator $\|.\|$ represents the evaluation of the Euclidean distance.

The access force that originates in the $\mathrm{UE_i}$ and acts on the $\mathrm{FlyBS_j}$ is formulated as:

$$\overrightarrow{F_{j,i}^{(A)}} = \frac{p_i}{P_{FlyBS}} \frac{\mathbf{r_{UE_i}} - \mathbf{r_{FlyBS_j}}}{\left\|\mathbf{r_{UE_i}} - \mathbf{r_{FlyBS_j}}\right\|} \tag{2.16}$$

where p_i refers to the power budget allocated at the $\mathrm{FlyBS_j}$ for the $\mathrm{UE_i}$. P_{FlyBS} represents the total power budget at the $\mathrm{FlyBS_j}$ and $\mathbf{r_{UE_i}}$ stands for the coordinates of the $\mathrm{UE_i}$ in the reference cell.

Altogether, the overall virtual force applied to the $\mathrm{FlyBS_j}$ is then defined as the sum of all virtual forces that act on the $\mathrm{FlyBS_j}$. This is expressed by the following formula:

$$\overrightarrow{F_j} = \overrightarrow{F_{j,b}^{B}} + \sum_{i \in \mathbb{U}_j} \overrightarrow{F_{j,i}^{A}} \tag{2.17}$$

where $\mathbb{U}_j$ represents the list of the UEs associated to the $\mathrm{FlyBS_j}$.

The vector of the overall virtual force expressed in (2.17) determines the direction in which the FlyBS shall be shifted in order to reach an equilibrium where all the forces acting on the FlyBS are balanced and where thus the overall virtual force is equal to 0.

2.4 Modified K-means Clustering

The K-means clustering algorithm is an algorithm that groups a set of analysed objects into K clusters by iteratively minimizing the distances between the objects and the cluster centroids by repositioning the cluster centroids and then associating the objects to the closest cluster centroid. However, when applying this algorithm to the scenarios where both the FlyBSs and the BS are deployed, the K-means algorithm introduces a major defect. In each iteration, the positions of the cluster centroid are recalculated and updated. This is applicable for the clusters that are served only by the FlyBSs, but as the position of the BS should remain unchanged, the K-means algorithm tends to become less accurate because the BS cannot follow the cluster centroid position. A possible solution for this defect could be removing the BS from the clustering algorithm. This would lead to situations in which all the UE would be connected to the network via the relay links. All UEs would be then associated to the FlyBSs. However, this would result in severe overloading of the FlyBSs, Thus, this solution is not desired.

The work [12] proposes a modified K-means algorithm that addresses the above-mentioned defect. This algorithm is, with minor adjustments, integrated in the proposed final framework presented in this thesis. Alongside considering conventional clusters whose centroid positions are updated in every iteration, this algorithm incorporates an additional cluster around the static centroid in the centre of the reference cell where the BS is positioned whose position remains unchanged throughout the whole algorithm execution.

To evaluate the distance between the object in the reference cell for the purposes of the modified K-means algorithm, the Euclidean distance is calculated. The distance between the $\mathrm{UE_i}$ and the BS is given as follows:

$$d_{i,b} = \|\mathbf{r_{UE_i}} - \mathbf{r_{BS}}\| \tag{2.18}$$

where the operator $\|.\|$ stands for the calculation of the Euclidean distance. The distances between the UE and the FlyBSs are calculated analogically with the same principle.

The positions of the cluster centroid are recalculated and updated in every iteration of the algorithm. The updated position of the cluster centroid k $\mathbf{r_k}$ is determined by this formula:

$$\mathbf{r_k} = \frac{\sum_{i=1}^{N_{UE_k}} \mathbf{r_{UE_i}}}{N_{UE,k}} \tag{2.19}$$

where $N_{UE,k}$ denotes the number of the UEs in the cluster k and $\overrightarrow{r_{UE_i}}$ denotes the position coordinates of the $\mathrm{UE_i}$.

The modified K-means algorithm is described in Algorithm 1. As the input parameters, the algorithm requires the number of FlyBSs in the reference cell N_{FlyBS} and the positions of the UEs $\mathbb{R}_{UE}$. Firstly, the number of clusters K is derived from the value of N_{FlyBS} in the line 1. The following step is to determine the centroid positions for every cluster that is served by one of the FlyBSs and save them to the set *Centroids*. This is done by electing a unique random position from $\mathbb{R}_{UE}$ for each centroid (lines 2 to 4). The position of the static centroid of the BS is then also saved to the set *Centroids* in the line 5.

After determining the initial centroid positions, the iterative part of the algorithm follows. This is described by the while loop (lines 7 to 23). Each iteration of the while loop is initiated with calculating the distances between the UE and each centroid (lines 9 to 12) and associating the UE to the closest centroid based on the calculated distances via (2.18) (line 13). For this purpose, the association set $\mathbb{A}$ is defined. This procedure is executed for each UE from the set $\mathbb{U}$ in the lines 8 to 14. When all the UE associations are updated, but no UE reassociation occurred, the association set $\mathbb{A}$ is considered final and the algorithm is terminated (line 16). If at least one reassociation occurred, the positions of the centroids are updated via (2.19) in the line 19 and algorithm's while loop starts from the beginning with a new iteration. After the initial clustering algorithm has ended, the UEs are associated according to the final set of associations (lines 24 to 30).

Algorithm 1 Modified K-means algortihm

```
Require: $N_{FlyBS}, \mathbb{R}_{UE}$
 1: Number of clusters $K \leftarrow N_{FlyBS} + 1$
 2: for each $k$ = 1:$N_{FlyBS}$ do
 3:     $Centroids(k) \leftarrow$ position of randomly elected UE from $\mathbb{R}_{UE}$
 4: end for
 5: $Centroids(K) \leftarrow$ position of the BS
 6: $\mathbb{A}_{prev} \leftarrow 0$
 7: while true do
 8:     for each $UE$ in $\mathbb{U}$ do
 9:         $d \leftarrow$ vector for storing distances between UE and FlyBS or BS
10:         for each $k$ in 1:$K$ do
11:             $d_k \leftarrow$distance between UE and FlyBS or BS via (2.18)
12:         end for
13:         $\mathbb{A}(UE) \leftarrow argmin(d)$
14:     end for
15:     if $\mathbb{A} = \mathbb{A}_{prev}$ then
16:         break
17:     else
18:         for each $k$ in 1:$K$ do
19:             $Centroids(k) \leftarrow$ new centroid position via (2.19)
20:         end for
21:         $\mathbb{A}_{prev} \leftarrow \mathbb{A}$
22:     end if
23: end while
24: for each $UE$ in $\mathbb{U}$ do
25:     if $\mathbb{A}(UE) = K$ then
26:         Associate UE with the BS
27:     else
28:         Associate UE with the FlyBS$_\mathrm{j}$ where $j = \mathbb{A}(UE)$
29:     end if
30: end for
```

2.5 Deep Neural Network Architecture

As thoroughly explained in [17], fundamental elements of the architecture of the DNNs are neurons. The neurons are computation blocks that receive inputs and based on their inner structure, the neurons produce an output. In Figure 2.3, a model of the inner structure of the neuron n_1 is presented. A set of N_{in} inputs from a previous layer is represented as $\mathbb{I} = \{in_1,\ldots,\ in_{N_{in}}\}$, a set of N_w connection weights is represented as $\mathbb{W} = \{w_1,\ldots,\ w_{N_w}\}$. The objective of the summation function is to bind the neuron inputs with the weights of their connections and calculate their sum which later serves as the input to the activation function. The letter b symbolizes the bias that is an additional parameter that helps to adapt the summation function output to the activation function [17]. Finally, the activation function is generally a non-linear mathematical function that determines the scalar output of the neuron based on its input from the summation function [18], [19].

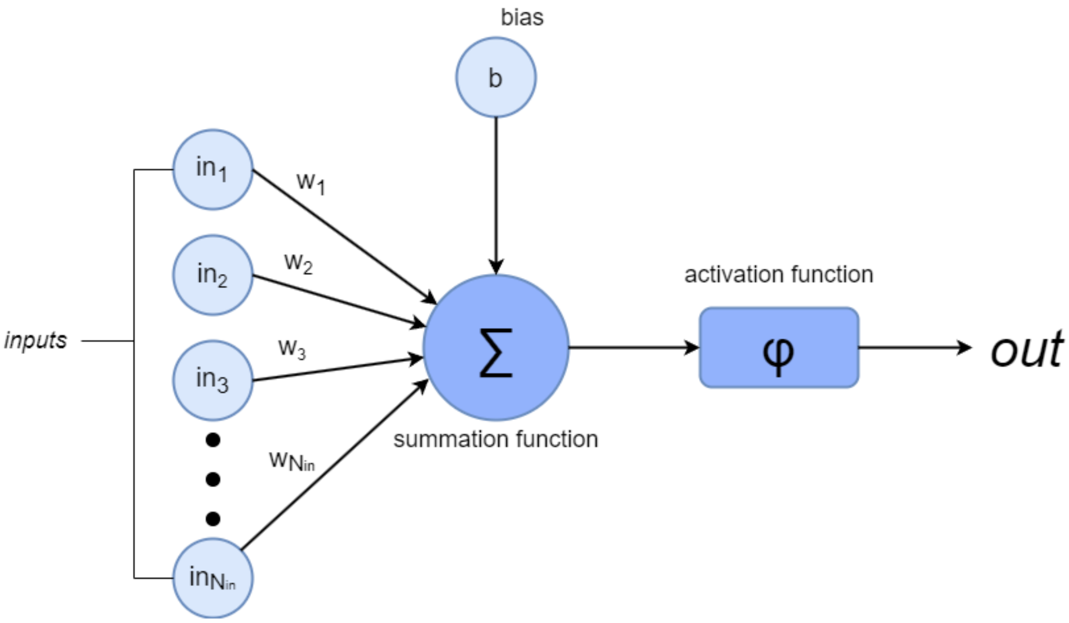


Figure 2.3: Inner structure of the neuron.

The output of the neuron is then calculated by the following formula [17]:

$$out = \varphi(\sum_{i=1}^{N_{in}} w_i \cdot in_i + b) \tag{2.20}$$

where out is the neuron output, N_{in} represents the number of the inputs, w_i stand for the i-th weight from the set of weights $\mathbb{W}$, in_i is the i-th input from the set of inputs $\mathbb{I}$ and b is the value of the bias. Finally, φ is represents the activation function.

The neurons are typically organized into layers, including one input layer L_{IN}, N_{hidden} hidden layers stored in the set $\mathbb{L}_{HIDDEN} = \{h_1,\ldots,h_{N_{hidden}}\}$ and one output layer L_{OUT}

that produces the final network output of the DNN, with each layer consisting of a different number of neurons. The neurons from neighbouring layers are interconnected via the weighted connections described above as the set $\mathbb{W}$. Figure 2.4 illustrates a general example of the DNN. The input layer L_{IN} consists of N_{input} neurons, where N_{input} is equal to the number of the input parameters, so called DNN predictors, that are processed by the DNN. The input layer is then connected to the first hidden layer h_1. The hidden layers are gradually interconnected and the last hidden layer $h_{N_{hidden}}$ is then connected to the output layer L_{OUT}. Finally, the output layer produces the output Y_{DNN}.

Also note the neuron n_1, whose structure was described in Figure 2.3, is deployed in Figure 2.4 as an element of the DNN.

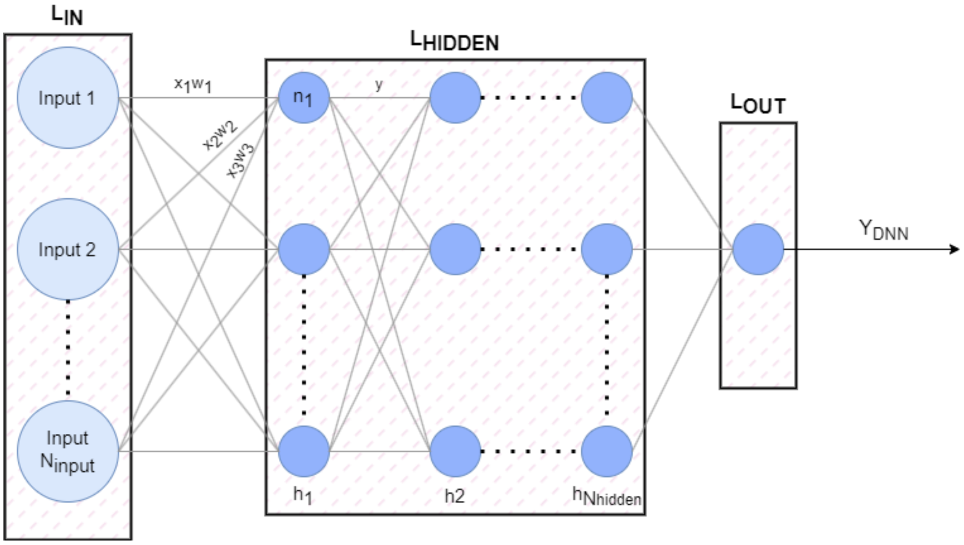


Figure 2.4: Inner structure of the DNN.

The DNN is trained via a process of backpropagation. The backpropagation involves taking an error that occurred during the final output prediction and sending information about this error backwards through all the layers of the DNN. Based on the backpropagation, the connection weight values are adjusted in order to reduce the error for the next predictions [17], [19].

Chapter 3

Problem Formulation

The main objective of this thesis is to propose a solution that maximizes the total channel capacity in the reference cell, in which one or more FlyBSs are deployed. In other words, the objective is not to maximize the channel capacity offered to a specific UE, but to find a solution that increases the overall sum of the channel capacities for all UEs located in the reference cell. This objective is addressed by two principal approaches:

- optimizing the set of the positions of the FlyBSs $\mathbb{R}_{FlyBS}$ in order to increase the channel capacity mainly for the UEs connected to the network via the relay links,
- optimizing the set of associations of the UEs $\mathbb{A}$ in order to better balance the loads of the FlyBSs and the BS.

Combining the above-mentioned approaches into a framework should improve the total channel capacity in the system model and thus result in the improved overall performance of the network. The objective of this thesis is then mathematically formulated as:

$$\mathbb{A}, \mathbb{R}_{FlyBS} = \underset{\mathbb{A}, \mathbb{R}_{FlyBS}}{\operatorname{argmax}} (\sum_{i=1}^{N_{UE}} c_i) \tag{3.1}$$

$$\{x_{min}, y_{min}, z_{min}\} \leq \mathbf{r_{FlyBS}} \leq \{x_{max}, y_{max}, z_{max}\}, \forall FlyBS \in \mathbb{F} \tag{3.1a}$$

$$\sum \mathbf{a_{UE}} = 1, \forall UE \in \mathbb{U} \tag{3.1b}$$

$$a_{FlyBS_{BS}} = 1 \land a_{FlyBS_{FlyBS}} = 0, \forall FlyBS \in \mathbb{F} \tag{3.1c}$$

where c_i is the channel capacity for the $\mathrm{UE_i}$ and N_{UE} represents the number of the UEs in the reference cell. The constraint (3.1a) states that each FlyBS must be positioned within the reference cell while the constraint (3.1b) restricts all the UEs to be associated with either only one FlyBS or the BS. The constraint (3.1c) expresses that each FlyBS is directly associated to the BS which means that there are not any multihop relay links in the reference cell.

Chapter 4

Proposed Solution

In this chapter, the proposed solution to the problem formulated in Chapter 3 is presented. The solution is focused on the two principal aspects: the optimization of the positioning of the FlyBSs and the optimization of the association of the UEs to the FlyBSs and potentially directly to the serving BS.

When addressing a complex problem, such as the mentioned optimization of the FlyBS positioning and the UE association, a mathematical description of an optimal solution could be overly complicated or it could be impossible to describe such solution mathematically. The DNNs are capable of processing great amount input data, discover hidden relations among input data and therefore estimate output values for complex problems that cannot be analytically derived. For that reason, to find solutions that provide satisfactory results for the mentioned positioning and associaiton problems, the characteristics of the DNNs indicate that the DNNs are a promising approach to estimate the network parameters and relations between the objects in the reference cell with which the DNNs are capable of predicting an optmized UE association scheme and an optimized positioning of FlyBSs policy. Therefore, this thesis utilizes the DNNs to address the problem formulated in Chapter 3.

This chapter is organized as follows. First, the FlyBS positioning subproblem is addressed. The solution is based on determining optimized coordinates for the deployed FlyBSs utilizing a system of DNNs. Then, the UE association subproblem solution is described in detail. The subproblem the UE association is also tackled using a DNN to predict an optimized UE association scheme. The solutions for both the positioning and the association subproblems are then combined into a framework that provides a joint solution to the problem of maximizing the overall channel capacity.

4.1 Positioning of Flying Base Stations

One of the two principal approaches addressing the main objective of maximizing overall capacity of the UEs in the reference cell is the optimization of the positions of the deployed FlyBSs. In the first part of this section, a concept of applying the virtual shift forces to reposition the FlyBSs is introduced in order to shift the FlyBSs to an optimized location. This is followed by the description of the structure of the DNNs which are trained to determine the optimized locations of the FlyBSs in the reference cell based on the concept of the mentioned virtual forces.

4.1.1 Shift Vector

Section 2.3 defines the system of the virtual backhaul and access forces that both act on the FlyBSs. The access virtual force is described by the formula (2.16) which is based on a ratio between the power allocated towards the UE from the FlyBS and the total power budget of the FlyBS. However, as the power budget at the $\mathrm{FlyBS_j}$ is distributed equally among all the associated UEs, the equation (2.16) shall be simplified as:

$$\frac{p_I}{P_F} = \frac{1}{N_{UE_{FlyBS_j}}} \tag{4.1a}$$

$$\overrightarrow{F_{j,i}^{(A)}} = \frac{1}{N_{UE_{FlyBS_j}}} \frac{\mathbf{r_{UE_i}} - \mathbf{r_{FlyBS_j}}}{\left\|\mathbf{r_{UE_i}} - \mathbf{r_{FlyBS_j}}\right\|} \tag{4.1b}$$

where $N_{UE_{FlyBS_j}}$ stands for the number of the associated UEs at the $\mathrm{FlyBS_j}$.

The FlyBS shift vector is determined by the overall virtual force vector calculated by (2.17) and represents the magnitude of the shift vector and the direction which the FlyBS is shifted in to reach the force equilibrium. However, taking into account that all virtual attraction forces are normalized, the magnitude of each virtual force $\overrightarrow{F_{j,i}^{(A)}}$ and $\overrightarrow{F_{j,b}^{(B)}}$ is lower than or equal to 1. Therefore, the magnitude of the overall virtual force for the $\mathrm{FlyBS_j}$ cannot be greater than the number of associated UEs to the $\mathrm{FlyBS_j}$ $N_{UE_{FlyBS_j}} + 1$. This is described in the following formula:

$$\forall \left\|\overrightarrow{F_{j,i}^{(A)}}\right\| \leq 1 \tag{4.2a}$$

$$\forall \left\|\overrightarrow{F_{j,b}^{(B)}}\right\| \leq 1 \tag{4.2b}$$

$$\left\|\overrightarrow{F_j}\right\| = \left\|\overrightarrow{F_{j,b}^{(B)}}\right\| + \sum_{i \in \mathbb{U}} \left\|\overrightarrow{F_{j,i}^{(A)}}\right\| \leq 1 + N_{UE_{FlyBS_j}} \tag{4.2c}$$

The equation (4.3) reflects the fact that the magnitude of the overall vector force has generally a significantly lower value of its magnitude in comparison to the cell size and therefore it would not be efficient to utilize the overall force vector as the FlyBS shift vector. For the mentioned reason, the positioning step magnitude factor is introduced. This parameter is used for scaling of the magnitude of the shift vector. The resulting shift vector applied to the $\mathrm{FlyBS_j}$ is then defined as:

$$\overrightarrow{v_j} = \alpha\overrightarrow{F_j} \tag{4.3}$$

where $\overrightarrow{v_j}$represents the shift vector for the $\mathrm{FlyBS_j}$. As the FlyBS is allowed to move in all three dimensions, the shift vector $\overrightarrow{v_j}$ is defined as $\overrightarrow{v_j} = \{v_1, v_2, v_3\}$. The symbol α stands for the positioning step magnitude vector and $\overrightarrow{F_j}$ is the overall virtual force vector.

4.1.2 Positioning Deep Neural Networks

As the virtual forces are dependent on the positions of all the objects in the reference cell, this solution is mostly suitable when the UEs provide their locations to the network. However, in cases when the locations of the UEs remain unknown, it is impossible to calculate the virtual forces with a tolerable accuracy. Therefore, it is necessary that the problem is circumvented by exploitation of other available information that could be gathered from the network, such as gains between the objects in the network, average capacities that the FlyBSs are able to provide to their associated UEs, numbers of the associated UEs to the FlyBSs etc. This work proposes a system of the DNNs to estimate the resulting shift vector when the locations of the UEs are not known. The trained system of the DNNs is stored in each FlyBS and controls the positioning of each FlyBS.

The aim of the proposed system of the positioning DNNs is to predict the shift vector in whose direction the FlyBS shall be shifted in order to optimize its position and therefore to increase the overall capacity of the network. Also, the system of the DNNs evaluates potential benefits of repositioning the FlyBS according to the estimated shift vector in terms of overall capacity increase. Hence, the proposed solution is composed of two DNNs while each of them has a different function. As the coordinates of the shift vectors are continuous values, predicting the shift vector can be seen as a regression problem and is, thus, addressed by the regression positioning DNN. On the other hand, when it comes to the shift vector evaluation, the binary classification positioning DNN is utilized to either approve or reject the predicted shift vector. The scheme of the introduced system of DNNs and their interconnection is presented in Figure 4.1.

When estimating the shift vector for the $\mathrm{FlyBS_j}$, the regression positioning DNN is utilized in the first step. The regression positioning DNN receives the input data, processes it and as the output, it estimates the shift vector $\overrightarrow{v_j}$. Then, the coordinates of $\overrightarrow{v_j}$ among other input data are passed to the classification positioning DNN that is trained to predict whether the repositioning of the $\mathrm{FlyBS_j}$ according to the predicted shift vector $\overrightarrow{v_j}$ increases or decreases the overall capacity of the network. If the classification positioning DNN estimates that the repositioning of the $\mathrm{FlyBS_j}$ positively impacts the overall cell capacity, it approves the shift vector $\overrightarrow{v_j}$ by setting the output of the classification positioning DNN Y_{clas} to 1 and the $\mathrm{FlyBS_j}$ is then repositioned accordingly. In the opposite case, the shift vector is rejected and the $\mathrm{FlyBS_j}$ remains in the same position as Y_{clas} is set to 0.

To enhance the performance of the system, the positioning DNNs utilize a method of recursion. As the coordinates of the shift vector are predicted by the regression positioning DNN, the shift vector is then stored and at the moment of the subsequent shift vector estimation, the previous shift vector is utilized as an input to the regression positioning DNN in order to provide a more accurate shift vector estimation. This information about their previous decision that is provided to the regression positioning DNN tends to enhance the overall performance of the positioning DNNs.

Both positioning DNNs are trained with methods of supervised learning and are trained based on offline-learning principles. The main benefit of offline learning is that no online real-world network measurements are required because all necessary data for training procedures are obtained from offline simulations. To attain objectives of the offline supervised training, simulation scenarios in the network model have been executed in order to obtain samples that together form training datasets. Each sample contains all necessary information about the predictors and the targets for the DNNs.

The targets are predicted by the DNNs as the output values. The regression positioning DNN targets are the coordinates of the estimated shift vector $\overrightarrow{v_j}$, while the target of the classification positioning DNN is a binary value Y_{clas} which either approves or rejects the estimated shift vector. This results in the output of the system of the DNNs $Y_{positioning}$ being either the approved shift vector $\overrightarrow{v_j}$ in case $Y_{clas} = 1$ or the empty vector $\{0, 0, 0\}$ in case of $Y_{clas} = 0$.

As explained above, when the UE locations are unknown, the aim of the DNN is to utilize different network parameters to predict the optimized FlyBS positions. Therefore, as an alternative to the UE locations, the set of predictors $\mathbf{Pred_{positioning}}$ processed by the positioning DNNs is defined. The set $\mathbf{Pred_{positioning}}$ contains:

- the positions of the FlyBSs in the reference cell to acquire a better overview of the distribution of the FlyBSs in the reference cell space,
- the gains between the FlyBSs to substitute the parameters of the distances between the FlyBSs,

- the gains between the FlyBSs and the BS,in the similarly as the previous parameters, to substitute the parameters of the distances between the FlyBSs and the BS,
- the numbers of the UEs associated to each FlyBSs and the BS to estimate better to loads of each FlyBSs and the BS,
- the average capacities of the UEs at each FlyBS to estimate better the loads of each FlyBS and the BS,
- interference experienced by the FlyBSs to acquire better understanding of the impacts of interference on the performance of the FlyBSs,
- the allocated power from the serving BS towards each FlyBS to better understand the relations between the BS and the FlyBSs in terms of power allocation,
- the coordinates of the shift vector utilized in the previous utilization of the system of the DNNs as the recursion method input. The DNNs shall predict a more accurate shift vector as the DNNs are provided values of a previous shift vector.

The number of the FlyBSs in the reference cell is generally not fixed and is different in different deployment scenarios. The inconstinency in the number of the FlyBSs generates in a non-constant number of the predictors for the positioning DNNs. However, the positioning DNNs are not designed to accept a varying number of the predictors. Therefore, for the sake of simplifying the complexity of the positioning subproblem, the number of the FlyBSs considered by the positioning DNNs is fixed to 3 in order to secure a constant number of the predictors for the positioning DNNs. In cases there are less than 3 FlyBSs in the reference cell, the values of the predictors that require information from the absent FlyBSs are set to 0. In cases in which there are more than 3 FlyBSs in the reference cell, the FlyBSs considered by the DNNs are determined by the gains between the FlyBS that is to be repositioned and the other FlyBSs. In particular, the FlyBS that is to be repositioned and two FlyBSs with the largest gains towards the FlyBS that is to be repositioned belong among the considered FlyBSs. Information from the other FlyBSs is not processed by the DNNs. If more FlyBSs were considered by the DNNs, the number of the predictors would grow exponentially, which would significantly increase the complexity of the DNNs and would result in overly long training times of the positioning DNNs. Thus, the idea of considering only 3 FlyBSs is applied.

Both positioning DNNs utilized in the proposed framework are designed to have fully-connected layers. As the activation function for the hidden layers, the rectified linear unit (ReLU) activation function is utilized. However, as the positioning DNNs have different types of outputs, more specifically, the regression positioning DNNs is supposed to produce

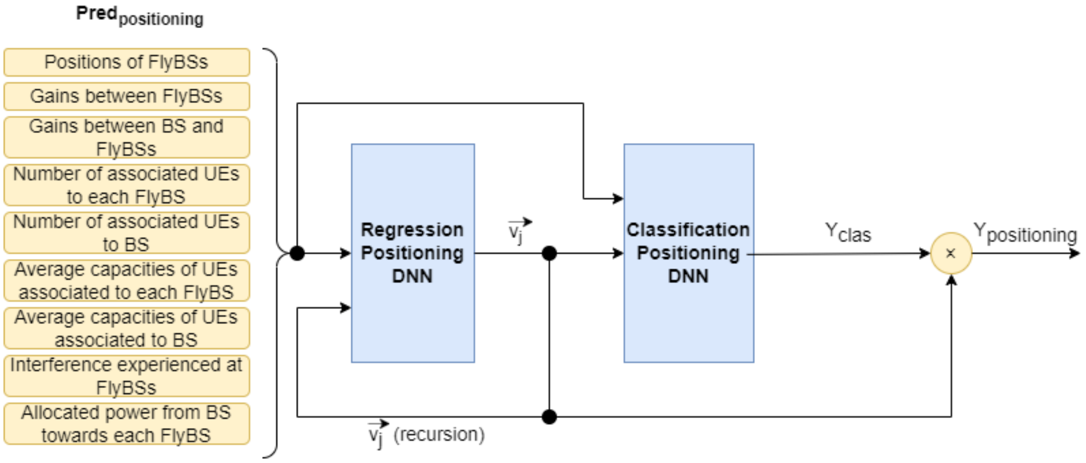

Pred$_{positioning}$
Positions of FlyBSs
Gains between FlyBSs
Gains between BS and FlyBSs
Number of associated UEs to each FlyBS
Number of associated UEs to BS
Average capacities of UEs associated to each FlyBS
Average capacities of UEs associated to BS
Interference experienced at FlyBSs
Allocated power from BS towards each FlyBS
Regression Positioning DNN
$\vec{v_j}$
Classification Positioning DNN
Y$_{clas}$
×
Y$_{positioning}$
$\vec{v_j}$ (recursion)

Figure 4.1: The scheme of the system of the positioning DNNs.

continuous values and the classification positioning DNNs is supposed to produce binary values, different activation functions are deployed for the output layers of the positioning DNNs. The regression positioning DNN utilizes the linear activation function for its output layers unlike the classification positioning DNN that uses the sigmoid activation function. Both the regression positioning DNN and the classification DNN are composed of 6 hidden layers with 128, 64, 32, 16, 8 and 4 neurons. The output layer of the positioning DNN is followed by the output layer with 3 neurons each predicting one coordinate of the shift vector $\vec{v_j}$. The output layer of the classification positioning DNN, on the other hand, consists of a single neuron producing the binary value Y_{clas}.

4.2 User Association

In this section, the principal concept of the UE association is described. The main objective of the UE association is to maximize the overall channel capacity in the reference cell. As the performance of the UE association is dependent on the positions of the FlyBSs and the positions of the FlyBSs are changed throughout the optimization process, the proposed concept consists of two phases. The first phase is the initial UE clustering based on the distances between the UEs and the FlyBSs and the BS. The clustering process is not optimal and is only utilized for the initial association. After the initial association procedure, the DNN is utilized to predict the UE associations forcing the UEs to potentially reassociate to one of the FlyBSs or the BS in order to increase the overall channel capacity.

4.2.1 Initial Association

The objective of the initial association is to determine groups of the UEs that will be associated to the same serving station, either the BS or one of the FlyBSs, in the reference cell. To simplify the problem of the initial UE association procedure, the positions of the UEs in the reference cell are considered known to the network for the phase of the initial association and therefore the modified K-means clustering presented in Algorithm 1 is used. For the reassociation phase described in Section 4.2.2, the coordinates of the UEs remain unknown in order to simulate real-world scenarios more accurately.

Although the initial association scheme determined by the modified K-means clustering algorithm provides a reasonable distribution of the UEs among the FlyBSs, it is not considered optimal. The association scheme takes into consideration only the distances between the objects omitting other impacting parameters, such as the loads of the FlyBSs or interference. The initial clustering algorithm associates the majority of the UEs to the FlyBSs which generally results in overloading the FlyBSs causing the FlyBSs not being capable of providing satisfactory channel capacities to the UEs. Moreover, the initial clustering algorithm fully relies on the knowledge of the UE locations, thus it is not suitable for scenarios in which the UE locations are not available to the network.

4.2.2 Reassocation Deep Neural Networks

The initial association algorithm utilizes information about the locations of the UEs in order to perform the initial association. However, to optimize the association of the UEs after the initial association, the proposed solution considers the UE locations unknown and therefore works without any information about the coordinates of the UEs. When the locations of the UEs are not known to the network, a promising option to substitute information about the UE locations is to utilize other information known to the network in a combination with the DNN. Hence, the proposed solution of the association subproblem is to utilize the DNN to find relations between available information and to use these relations to optimize the UE association procedure in the network.

Unlike in the solution of the positioning subproblem, only one DNN is trained to optimize the UE association. As the UE association is described by discrete values – each FlyBS has its own index, the values are not continuous and therefore the UE association is perceived as a classification problem. For this reason, the multiclass-classification DNN is trained to predict the optimal association of each UE in the network. The scheme of the association classification DNN is presented in Figure 4.2.

Once the association classification DNN receives the input data, the input data is processed and as the output $Y_{association}$, the association DNN makes a prediction which

FlyBS or the BS should be the UE associated to in order to secure the highest overall channel capacity of the network. The association DNN can predict either reassociation to another FlyBS or the BS or it can also predict that in terms of overall network capacity the best option is to keep the UE association unchanged. Note that it does not necessarily mean that the DNN predicts the association that secures the highest channel capacity for the UE as it is not the main goal of the DNN.

In the similar way as the positioning DNNs, the association classification DNN is trained with the methods of supervised learning and is trained based on the offline-learning principles. Offline simulations in the network model have been executed in order to obtain training datasets. For the association classification DNN, the target $Y_{association}$ is the index of the FlyBS or the BS that the UE will be associated to. As considering that the UE locations are unavailable for the network for this phase of the positioning subproblem solution, the set of predictors $\mathbf{Pred_{association}}$ is defined as:

- the number of the UEs associated to each FlyBSs and the BS to estimate the loads of each FlyBSs and the BS,
- the average capacities of the UEs at each FlyBS and the BS to estimate better loads of each FlyBS and the BS in a similar way as the above-mentioned parameters,
- the gains between the FlyBSs, gains between the BS and the FlyBSs to estimate better again the loads of each FlyBS and the BS,
- the gains between the UE and the FlyBSs and the BS to substitute the parameters of the distances to the FlyBSs and the positions of the FlyBSs,
- interference experienced by the FlyBSs to better understand the impacts of interference on the performance of the FlyBSs,
- the allocated power from the BS towards the FlyBSs to understand the relations between the BS and the FlyBSs in terms of power allocation in the reference cell,
- the index of the FlyBS or the BS that the UE is currently associated to in order to provide information about the current association setup.

As the number of FlyBSs in the network is variable, the number of FlyBSs that are considered by the DNN is reduced to 3 in a similar way as in the system of the positioning DNNs. In case of the association DNN, the considered FlyBS are the three FlyBSs with the largest gains towards the UE. If less than 3 FlyBSs are deployed in the reference cell, the predictors that require information from the absent FlyBSs are set to 0. In case that there are more than 3 FlyBSs in the reference cell, information from the FlyBSs that are not considered by the association DNN is discarded. If more FlyBSs are considered by the DNN, the number of the predictors grows exponentially, which significantly increases the complexity of the DNN which results in overly long training times of the positioning DNNs. Thus, the idea of considering only three FlyBSs is applied again.

The association DNN is composed of fully-connected layers. As the activation function for the hidden layers, the ReLU activation function is utilized. As the activation function for the output layer, the classification association DNN use the sigmoid activation function. The classification association DNN is composed of 5 hidden layers with 128, 64, 32, 16 and 8 neurons respectively which are followed by the output layer that is composed of 4 neurons predicting the discrete value target $Y_{association}$.

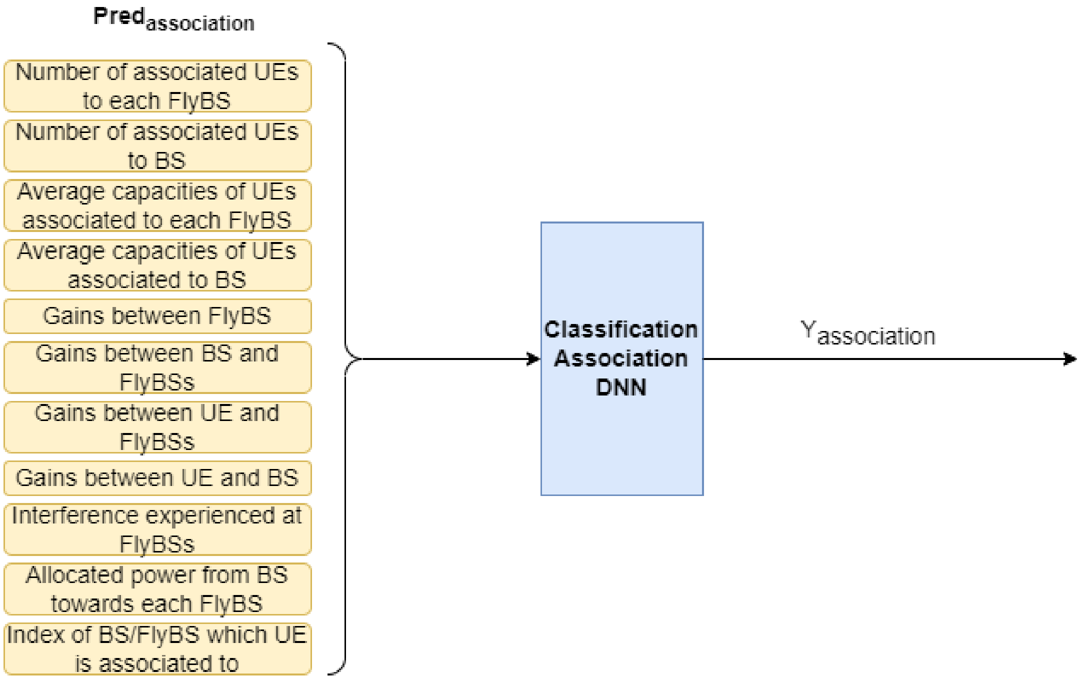

Pred association
Number of associated UEs to each FlyBS
Number of associated UEs to BS
Average capacities of UEs associated to each FlyBS
Average capacities of UEs associated to BS
Gains between FlyBS
Gains between BS and FlyBSs
Gains between UE and FlyBSs
Gains between UE and BS
Interference experienced at FlyBSs
Allocated power from BS towards each FlyBS
Index of BS/FlyBS which UE is associated to
Classification Association DNN
Y association

Figure 4.2: The scheme of the association DNN.

4.3 Final Framework

So far, both the positioning and the association subproblems have been addressed separately. The proposed final solution combines both concepts together creating a general framework that optimizes the overall channel capacity in the reference cell.

The final framework utilizes two DNNs for the positioning of the FlyBSs and one single DNN for the association of the UEs as described in Section 4.1 and Section 4.2. However, the DNNs do not operate separately but are interconnected and cooperate to enhance the overall performance of the framework. The aim of the DNN interconnection is to provide more input information to the DNNs that should lead to more accurate predictions and therefore more optimized association and positioning decisions. The proposed scheme of the interconnection of the DNNs is presented in Figure 4.3.

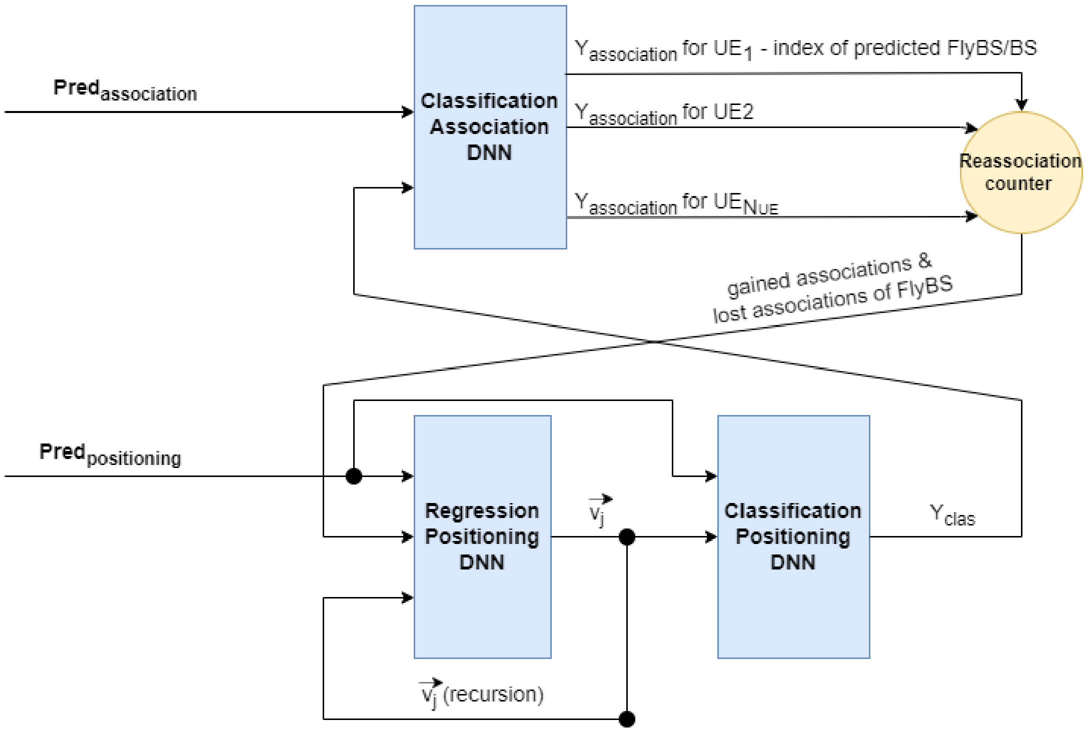

Pred association
Classification Association DNN
Y association for UE1 - index of predicted FlyBS/BS
Y association for UE2
Y association for UE NUE
Reassociation counter
gained associations & lost associations of FlyBS
Pred positioning
Regression Positioning DNN
v j
Classification Positioning DNN
Y clas
v j (recursion)

Figure 4.3: The scheme of the interconnection of the DNNs in the final framework.

The interconnection of the positioning DNNs and the cooperation between each other, the principle of the DNN recursion is described in detail in Section 4.1.2. Furthermore, the binary output ot the positioning classification DNN named Y_{clas} is also mapped to the input of the association DNN and serves as one of the predictors for the DNN. This predictor equals 0 if the FlyBS that the UE is connected to was not repositioned in the previous iteration of the algorithm. If the FlyBS was repositioned, the predictor is equal to 1. In case of the UE being associated with the BS, the value of this predictor is automatically set to 0. The objective of introducing this predictor is to provide information summarizing if the position of the FlyBS which the UE is associated to was changed or not to the association DNN. The repositioning of the FlyBSs changes the cell characteristics, such as interference, gains among the UEs and FlyBS etc., therefore it serves as a valid predictor for the association DNN.

In addition, the combined output of the association DNN is mapped to the input of the regression positioning DNN and serves as two predictors for the DNN. The predictors fetch information about the number of the UEs that were newly associated to the particular FlyBS and about the number of the UEs that were reassociated to another FlyBS or to the BS. In other words, the predictors carry information about the gained associations and the lost associations. Collecting of these predictors is denoted as *Reassociation counter* in Figure 4.3. These predictors should help to improve the accuracy of the positioning DNNs predictions as the number of the associated UEs and by extension, the change of this number greatly influences the final shift vector which the positioning DNNs aim to predict.

The final framework is described in Algorithm 2. As the first step (line 1), the initial clustering algorithm described in Algorithm 1 is executed determining the initial association which is stored to the set of associations $\mathbb{A}$. After that, each FlyBS is positioned to the assigned cluster centroid and their location is stored in the set of positions $\mathbb{R}_{FlyBS}$ (lines 2 to 5). Subsequently, the capacities of the UEs are evaluated and stored in the set of UE capacities $\mathbb{C}$ (line 6). This is followed by algorithm's main loop. Every iteration of the loop begins by storing the sets of the capacities $\mathbb{C}$ as $\mathbb{C}_{prev}$, the set of associations $\mathbb{A}$ as $\mathbb{A}_{prev}$ and the set of the FlyBS positions $\mathbb{R}_{FlyBS}$ as $\mathbb{R}_{FlyBS_{prev}}$ (lines 8 to 10) determined either at the beginning of the execution of the algorithm or possibly in the previous iteration of the algorithm. This is followed by the determination of the mean of the UE channel capacities in the line 11. Then, the UEs are sorted according to their channel capacities in the ascending way (from the UE with the lowest channel capacity to the UE with the greatest channel capacity) in the line 12. This is followed by the UE reassociation block (lines 13 to 20). For each UE, it is examined whether its channel capacity is lower than the average channel capacity (line 14). If so, the association DNN predicts the best association for the UE (line 15). If the predicted association differs from the actual association (line 16), the set of associations $\mathbb{A}$ is updated and the UE is reassociated accordingly (line 17). This block of the algorithm is repeated for each UE in the reference cell.

This condition of applying the DNN association process only to the UEs whose channel capacity is lower than the average channel capacity in the reference cell is introduced to the algorithm due to occasional errors of the association DNN when the association DNN does not predict a better association but predicts an association that actually decreases the overall channel capacity in the reference cell. In cases when a wrong prediction output is generated for the UEs with greater channel capacities, the channel capacity decrease tends to impact the overall channel capacity more than cases when a wrong prediction output is generated for the UEs with lower channel capacities. Therefore, the UEs with greater channel capacities are omitted from the reassociation process. The threshold of the average capacity in the reference cell that determines which UE undergoes the reassociation process was determined by the trial-error approach.

The UE reassociation block is followed by the FlyBS repositioning block (lines 21 to 28). For each FlyBS, the regression positioning DNN predicts the shift vector (line 22) which is evaluated by the classification repositioning DNN (line 23). If the classification

DNN predicts that the shift vector will increase the overall channel capacity in the reference cell (Y_{clas} equals 1), the FlyBS is repositioned according to the shift vector and the new position of the FlyBS is stored to the set of FlyBS positions (lines 24 to 26). This procedure is repeated for each FlyBS in the reference cell. After that, the capacities for the UEs are again evaluated (line 29) and are compared to the capacities obtained in the previous iteration of the while loop. If the sum of the newly obtained capacities is lower than or equals the capacities from the previous iteration, the algorithm is terminated (lines 30 and 31). As the output, the sets of UE associations $\mathbb{A}_{prev}$, FlyBS positions $\mathbb{R}_{FlyBS_{prev}}$ and the set of capacities $\mathbb{C}_{prev}$ from the previous iteration are returned.

Algorithm 2 Final Framework

```
 1: $\mathbb{A} \leftarrow$ Perform initial association via Algorithm 1
 2: **for** each $j = 1{:}N_{FlyBS}$ **do**
 3:     $\mathbf{r_j} \leftarrow Centroid(j)$
 4:     $\mathbb{R}_{FlyBS}(j) \leftarrow \mathbf{r_j}$
 5: **end for**
 6: $\mathbb{C} \leftarrow$ evaluate UE channel capacities
 7: **while** true **do**
 8:     $\mathbb{C}_{prev} \leftarrow \mathbb{C}$
 9:     $\mathbb{A}_{prev} \leftarrow \mathbb{A}$
10:     $\mathbb{R}_{FlyBSprev} \leftarrow \mathbb{R}_{FlyBS}$
11:     $C_{mean} \leftarrow$ calculate average UE capacity in the cell
12:     $\mathbb{U}_{sorted} \leftarrow$ sort UEs according to their channel capacity in the ascending way
13:     **for** each $UE$ in $\mathbb{U}_{sorted}$ **do**
14:         **if** $\mathbb{C} \leq C_{mean}$ **then**
15:             $Y_{association} \leftarrow$ perform association prediction
16:             **if** $\mathbb{A}(UE) \neq Y_{association}$ **then**
17:                 $\mathbb{A}(UE) \leftarrow Y_{association}$
18:             **end if**
19:         **end if**
20:     **end for**
21:     **for** each $FlyBS$ in $\mathbb{F}$ **do**
22:         Shift vector $\leftarrow$ perform shift vector prediction via regression positioning DNN
23:         $Y_{clas} \leftarrow$ perform shift vector evaluation via classification positioning DNN
24:         **if** $Y_{clas} == 1$ **then**
25:             $\mathbf{r_{FlyBS}} \leftarrow$ reposition the FlyBS according to the shift vector
26:             $\mathbb{R}_{FlyBS}(FlyBS) \leftarrow \mathbf{r_{FlyBS}}$
27:         **end if**
28:     **end for**
29:     $\mathbb{C} \leftarrow$ evaluate UE channel capacities
30:     **if** $\sum \mathbb{C} \leq \sum \mathbb{C}_{prev}$ **then**
31:         break;
32:     **end if**
33: **end while**
34: Return $\mathbb{A}_{prev}$, $\mathbb{R}_{FlyBS_{prev}}$ and $\mathbb{C}_{prev}$
```

Chapter 5

Performance Evaluation

In the first section of this chapter, the parameters of the simulation model are stated. This is followed by the description of the competitive schemes against which the proposed framework is compared. Finally, the performance evaluation metrics are defined.

5.1 Simulation Model Description

The system model described in Chapter 2 is designed in MATLAB. For the purposes of this thesis, the reference cell is square-shaped with dimension lengths of 1000 $\times$ 1000 m. The reference cell is served by 1 BS that is located in the cell centre. The reference cell is surrounded by 4 neighbouring cells. The neighbouring cells are introduced to the model to simulate inter-cell interference. Each of these cells is represented by 1 BS. The distance between the neighbouring BSs and the borders of the reference cell is 600 m.

The altitude of BS transceivers is fixed to 30 m. Similarly, the altitude of UE transceivers is fixed to 1.5 m. Unlike altitudes of the BS and the UE transceivers, the altitudes of FlyBSs are not fixed and change in the simulations. However, the altitudes of FlyBS must be higher than 10 m. This constraint is introduced in order to avoid possible collisions with obstactles in the cell and to avoid disturbing the mobile network users.

The power budget of the BS is defined as 30 dBm and the power budget of each FlyBS is defined as 23 dBm. The carrier frequency for communication among all the objects in the reference cell is 3.5 GHz with the bandwidth of 40 MHz. Also, for interference purposes and thermal noise simulations, the value of the noise spectral density is set to -174 dBm/Hz. The above-mentioned parameters are summarized in Table 5.1. Figure 5.1 shows the scheme of the system with the defined model parameters.

Table 5.1: The network model parameters

Parameter	Symbol	Value	Description
Cell dimension lengths	d_1, d_2	1 × 1 km	Lenghts of cell dimensions
BS height	z_{BS}	30 m	Altitude in which the BS transceiver is positioned
UE altitude	z_{UE}	1.5 m	Altitude in which the UE transceivers are estimated
FlyBS altitude constraint	z_{min}	10 m	The lowest altitude the FlyBS is allowed to be located at
BS power budget	P_{BS}	30 dBm	Total downlink transmission power available at the BS
FlyBS power budget	P_{FlyBS}	23 dBm	Total downlink transmission power available at the FlyBS
Carrier frequency	f	3.5 GHz	Central frequency of the communication band
Cell bandwidth	B	40 MHz	Total bandwidth of the reference cell
Noise spectral density	σ	-174 dBm/Hz	Noise spectral density calculated in dBm

In the simulated scenarios, the number of the FlyBSs in the reference cell is not fixed and ranges between 1 and 10 and the number of the UEs in the reference cell varies between 20 and 100. The value of the positioning step magnitude factor is 100. This factor is used for scaling of the repositioning shift vector defined in (4.3). Table 5.2 summarizes the simulation parameters.

Table 5.2: The simulation parameters

Parameter	Symbol	Value	Description
Number of UEs	N_{UE}	20 to 100	Total number of UEs located in the reference cell
Number of FlyBSs	N_{FlyBS}	1 to 10	Total number of FlyBSs deployed in the reference cell
Positioning step magnitude factor	α	100	Parameter that increases the FlyBS shift vector magnitude

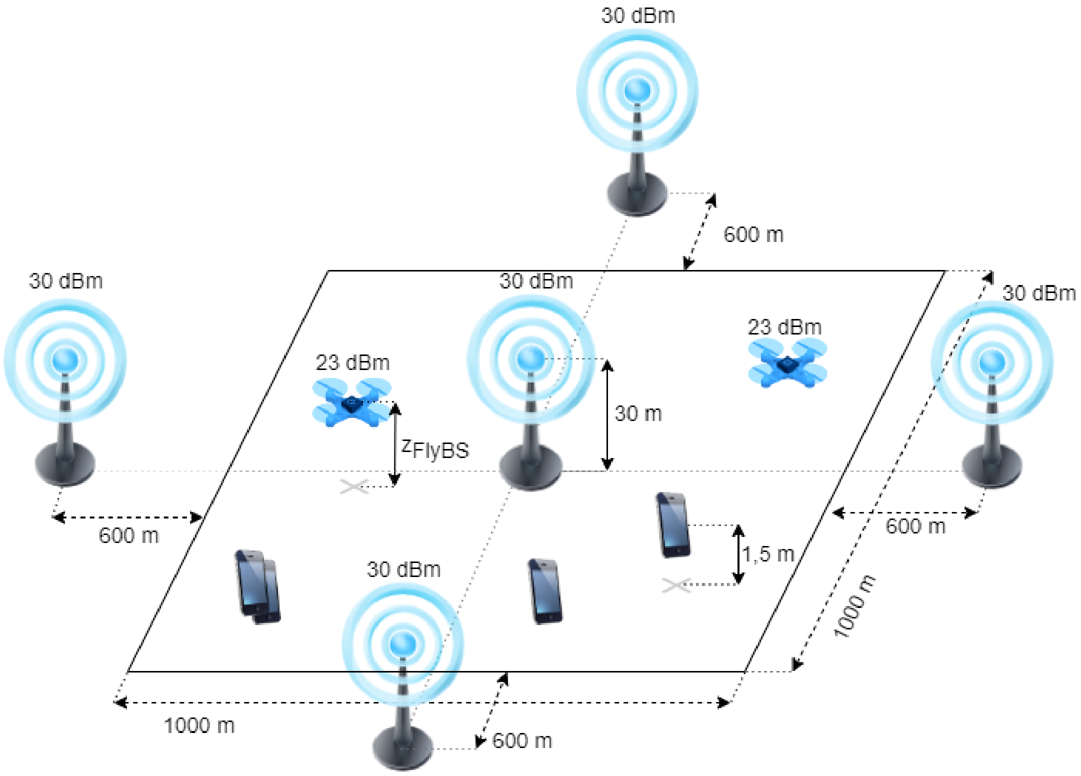


Figure 5.1: The parameters of the network model.

5.2 Competitive Schemes

With the aim of acquiring a better perspective of the performance of the proposed solution, the final proposed framework is compared to the following competitive schemes in the final simulations.

System of Separate DNNs

This competitive scheme utilizes the DNNs to optimize the positioning of the FlyBSs and the association of the UEs the reference cell in a similar manner as the proposed final framework. The scheme utilizes the regression positioning DNN to estimate the optimized shift vector and the association DNN whose objective is to predict the optimized UE association. But unlike the proposed final framework, this competitive scheme does not utilize advanced recursion methods, such as utilizing the shift vector estimated in the previous iteration of the algorithm as a predictor for the regression positioning DNN. Also, the classification positioning DNN is removed from this competitive scheme. Besides that, the association DNN and positioning DNN function in a separate way. In the final framework, the association DNN and the positioning DNN exchange information about the positioning status and the number of the lost and gained connections. However, as a consequence of the separation of the two domains in this scheme, the association DNN and the positioning DNN are not interconnected which means that they do not exchange any information.

K-means Clustering

In this competitive scheme based on the work presented in [20], the UEs are grouped into the clusters based on their distances from the centroids via the principles of K-means clustering described in Algorithm 1. Each cluster is served by a single FlyBS that is located in the cluster centroid with the exception of one static centroid that is located in the centre of the reference cell and is served by the BS. Although this competitive scheme distributes the UEs among the FlyBSs and the BS and the positions the FlyBS in the reference cell space are determined with reasonable logic, as the K-means algorithm does not consider other parameters influencing the performance of the cell such as power budgets, the scheme generally results in in the FlyBSs being overloaded.

Greedy Algorithm

The greedy algorithm scheme based on the algorithm presented in [21] aims to maximize the channel capacity for each UE individually. All the FlyBSs are first positioned to the centroids determined by the K-means clustering algorithm. Subsequently, the UEs individually analyse all the possible associations and select the association where the UE is provided the best channel capacity. This leads to the association decision logic being transferred from the network directly to the UEs.

However, the greedy algorithm presents various drawbacks. The process of the UE finding the association with the highest channel capacity is not governed by the network which has a better overview of the network association parameters, such as interference or how loaded the FlyBSs are. Also, each UE analyses all the possible associations which leads to the algorithm being much more time-consuming compared to the other competitive schemes. Therefore, utilizing the greedy algorithm applied to all UEs that consider every FlyBS in the reference cell as a potential association target is highly inefficient from perspective of the long computation times and the long convergence times. This leads to the fact that the greedy algorithm cannot be easily implemented in the real-world scenarios. Therefore, for the purposes of this thesis, 3 simplified versions of the greedy algorithm are presented:

- the greedy algorithm that is not applied to all the UEs in the reference cell, but only to 25 % of the UEs with the lowest channel capacity, considering all FlyBSs in the reference cell when making an association decision (denoted as Greedy25 - all FlyBSs),

- the greedy algorithm that is again applied only to 25 % of the UEs with the lowest channel capacity considering only 3 nearest FlyBSs in the reference cell when making an association decision (denoted as Greedy25 - 3 nearest FlyBSs),
- the greedy algorithm that is applied to all the UEs in the reference cell considering again 3 nearest FlyBSs when making an association decision (denoted as Greedy100 - 3 nearest FlyBSs).

Regarding the performance evaluation presented in Chapter 6, it is demonstrated that the greedy algorithms applied only to 25 % of the UEs with the lowest channel capacity deliver similar results regardless if all the FlyBS or only 3 nearest FlyBSs are considered in the reassociation process. It is then deduced that the number of the considered FlyBSs does not have a significant impact on the performance of the greedy algorithm as the FlyBSs that are near to the UEs tend to offer higher channel capacities compared to other distant FlyBSs.

On the other hand, the greedy algorithm that is applied to all the UEs in the reference cell reports a significantly better performance in terms of the overall channel capacity than the other greedy algorithms. This leads to the conclusion that the number of the UEs that the greedy algorithm is applied to significantly influences the performance of the greedy algorithms. However, the increasing number of the UEs that the algorithm is applied to rapidly increases the complexity and the convergence times. This leads to the greedy algorithms being practically unusable in the real-time scenarios, even if the greedy algorithm is applied to a fraction of the UEs in the reference cell.

5.3 Performance Metrics

In order to evaluate the performance of the proposed solution and the other competitive schemes, it is necessary to define the metrics that the performance evaluation will be based on. As the problem formulation defined in Chapter 3 states, the main objective of the proposed solution is to maximize the overall channel capacity in the reference cell. This is mathematically expressed as:

$$C_{total} = \sum_{i=1}^{N_{UE}} c_i \tag{5.1}$$

where C_{total} is the overall channel capacity in the reference cell, N_{UE} stands for the number of the UEs in the reference cell and c_i represents the channel capacity of the $\mathrm{UE_i}$.

What is also considered important, is how the channel capacities of the UEs vary in the reference cell. To evaluate the proposed solution and to compare it with the other competitive schemes, Jain's fairness index is introduced. Jain's fairness index is expressed by the following formula:

$$I_{Jain} = \frac{(\sum_{i=1}^{N_{UE}} c_i)^2}{N_{UE}\sum_{i=1}^{N_{UE}} c_i^2} = \frac{1}{1+\widehat{c_v}^2} = \frac{1}{1+(\frac{\sigma}{\mu})^2} \tag{5.2}$$

where I_{Jain} stands for Jain's fairness index, N_{UE} is the number of the UEs in the reference cell, c_i represents the capacity of the $\mathrm{UE_i}$, $\widehat{c_v}$ is the coefficient of variation, σ is the standard deviation of the set of the channel capacities and μ stands for the mean of the set of the channel capacities.

Also, to better understand the UE association policies, the performance metric showing the rate of the UEs in the reference cell associated to the FlyBSs is introduced. This is formulated as:

$$\varphi_{association} = \frac{N_{UE\_FlyBS}}{N_{UE}} \tag{5.3}$$

where $\varphi_{association}$ stands for the rate of the UEs associated to the FlyBSs, N_{UE_FlyBS} represents the number of the UE associated to the FlyBSs and N_{UE} is the number of the UEs in the reference cell.

Finally, the last metric that represents the average number of the UEs that are associated per FlyBS is defined. This metric is expressed by the following formula:

$$N_{UE/FlyBS} = \frac{N_{UE\_FlyBS}}{N_{FlyBS}} \tag{5.4}$$

where $N_{UE/FlyBS}$ stands for the average number of UEs associated to one FlyBS, N_{UE_FlyBS} represents the number of the UEs associated to the FlyBSs and N_{FlyBS} is the number of the FlyBSs in the reference cell.

Chapter 6

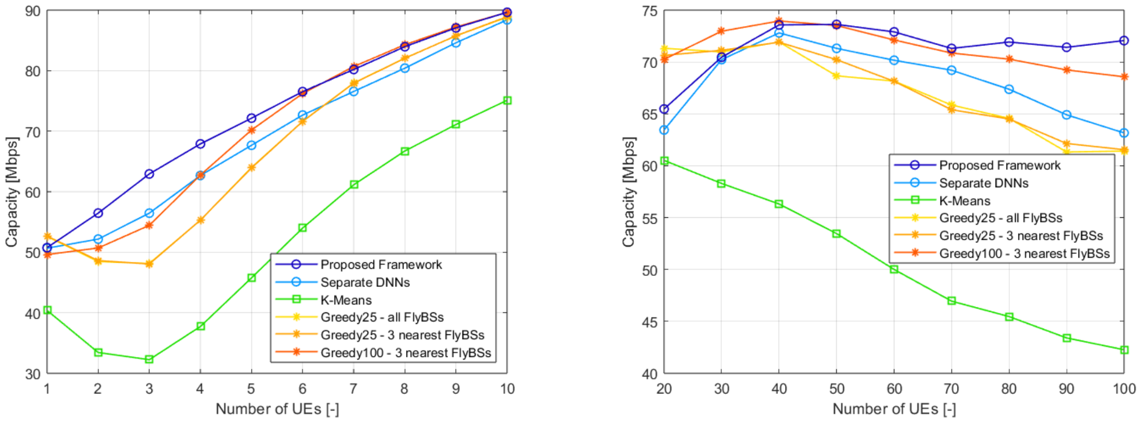
Simulation Results

In this chapter, the performance of the proposed solution framework is evaluated according to the defined performance metrics. The framework is also compared with the competitive schemes described in the previous chapter. To evaluate the performance of the proposed framework and to compare it with the competitive schemes, 100 different simulations were executed for each simulation scenario. The positions of the UEs were selected randomly at the beginning of each simulation and remained unchanged throughout the whole simulation.

6.1 Overall Channel Capacity

The main objective of the thesis is to maximize the sum of the channel capacities in the reference cell. Figure 6.1 displays this metric in two performance evaluation graphs. In Figure 6.1a, describing the overall channel capacity dependence on the number of the FlyBSs while the number of the UEs is fixed, the schemes utilizing the DNNs (the proposed framework and the system of separate DNNs) and the greedy algorithm applied to all the UEs report a gradual increase in the overall channel capacity.

On the other hand, the other competitive schemes (the greedy algorithms applied to 25% of the UEs and the K-means algorithm) report increases in their overall channel capacities only when $N_{FlyBS} > 3$. This anomaly is caused by the following sequence of reasons. The K-means clustering algorithm associates the UEs to the FlyBSs or the BS only based on the distances and does not take into account other parameters, such as how loaded the FlyBSs already are. Then, relatively many UEs are associated to the FlyBSs, and therefore the FlyBSs are able to allocate significantly low power to the UEs, which leads to the low channel capacities for the UEs associated to the FlyBSs caused by overloading of the FlyBSs. The access link, therefore, serves as a bottleneck. If there are two FlyBSs deployed in the reference cell, compared to the scenario with only one FlyBS,



(a) N_{FlyBS} dependence, $N_{UE} = 75$

(b) N_{UE} dependence, $N_{FlyBS} = 5$

Figure 6.1: Overall channel capacity depending on the number of FlyBSs and UEs.

there are noticeably more UEs associated to the FlyBSs, that is, there are generally more UEs in the network with a low channel capacity, and therefore the overall capacity is worse. However, as the number of the FlyBSs in the reference cell continues to increase, each FlyBS tends to have fewer associated UEs. The FlyBSs are then able to allocate more transmission power for each UE, and therefore the total capacity starts to increase. And as the greedy algorithms that are applied to 25% of the UEs utilize the K-means algortihm as their initial clustering algorithm, these greedy algorithms are also impacted by this anomaly. The greedy algorithm applied to 100% of the UEs does not report a similar behaviour because it tends to reassociate the majority of the UEs to the BS as observed in Section 6.3.

Also, note that there is no significant difference in the performances between the cases when the greedy algorithm applied to 25% of the UEs considers only 3 nearest FlyBSs and when all FlyBSs are considered as an reassociation option. Both algorithms report the identical performance, therefore their curves overlap in Figure 6.1a. This is caused by the fact that the UEs tend to reassociate to the FlyBSs that are nearby and not to the other distant FlyBSs. Therefore, thelimitation of considering only a limited number of the FlyBSs does not impact the overall performance of the algorithm while reducing the computation time and the complexity.

As illustrated in Figure 6.1a, the proposed framework generally outperforms the other competitive schemes in terms of the overall channel capacity maximization. The system of the separated DNNs is outperformed by up to 11%. The percentage difference generally decreases with the increasing number of the FlyBSs in the reference cell as the final framework and the system of the separated DNNs seem to converge to similar results. For $N_{FlyBS} \geq 6$, the greedy algorithm applied to all the UEs also reports very similar results as the final framework converging to similar values. For $N_{FlyBS} < 6$, the greedy algorithm applied to all the UEs is outperformed by the final framework by up to 16%. The greedy algorithms applied to 25% of the UEs are outperformed by the final framework by up to 31% for all cases, except for the case when $N_{FlyBS} = 1$. Finally, the K-means algorihm shows the lowest overall capacity for all simulation scenarios, being outperformed by the proposed framework by 19% - 95%.

Figure 6.1b illustrates how the overall channel capacity differs with the varying number of the UEs. The proposed final framework generally outperforms the other competitive schemes (for $N_{UE} \geq 40$) and is the only scheme that is capable of maintaning the value of the overall channel capacity with the increasing number of the UEs in the reference cell. All the greedy algorithms and the system of the separated DNNs report the highest values of the overall channel capacity for $N_{UE} = 40$. In cases with a greater number of the UEs in the reference cell, the mentioned schemes report gradual decreases in the overall channel capacity with the the greedy algorithm applied to all the UEs delivering generally better results than the other greedy algorithms and the system of the separated DNNs. The K-means algorithm reports the lowest values from all compared schemes for all simulation scenarios and also does not perform well in maintaining the overall channel capacity with the increased number of the UEs in the reference cell. This may be caused by the fact that with the increased number of the UEs, the FlyBSs associate too many UEs and therefore are not capable of allocating the sufficient power towards each UE which leads to insufficient channel capacities in the access links.

Both greedy algorithms applied 25% of the UEs again follow the same trend and only report minor differences in the overall channel capacity performance. It is also observed that the result curves of the final framework and the greedy algorithms applied to 25% of the UEs are not flattened. This could by caused by an insufficient number of the executed simulations.

All the greedy algorithms deliver higher overall channel capacity for the case when $N_{UE} = 20$ outperforming the proposed framework by 9%. However for the simulation cases when $N_{UE} \geq 50$, the final framework outperforms all the other competitive schemes. More specifically, the system of the separated DNNs is outperformed by up to 14%, the greedy algorithms applied to 25% of the UEs by 5%-17%, the greedy algorithm applied to all the UEs by up to 5% and the K-means clustering by 27% - 70%. The values of the percentages showing how much the final framework outperforms the other competitive schemes generally increase with the increasing number of the UEs in the reference cell.

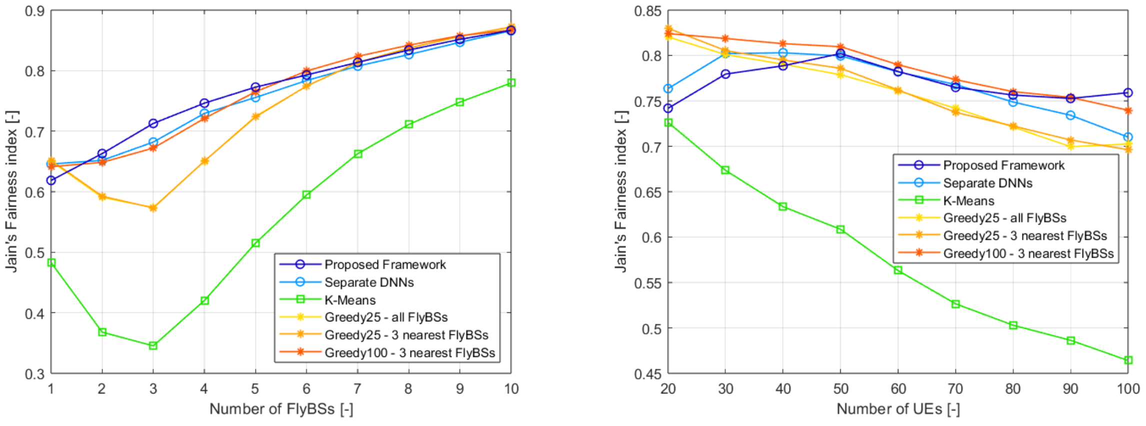
6.2 Jain's Fairness Index

In order to acquire a better overview of how the channel capacities in the reference cell are distributed, Jain's fairness index was defined. Figure 6.2 shows the performance evaluation of the proposed final framework and the competitive schemes based on mentioned Jain's fairness index.

In Figure 6.2a, describing the dependence of Jain's fairness index on the number of FlyBSs, the proposed final framework, the system of the separated DNNs and the greedy algorithm applied to all the UEs present a steady growth of the value of Jain's fairness index with the proposed framework presenting slightly better results between $N_{FlyBS} = 2$ and $N_{FlyBS} = 5$. For greater number of the FlyBSs, all the greedy algorithms converge to the similar values of Jain's fairness index as the final framework and the system of separated DNN while the K-means algorithm delivers significantly lower values of Jain's fairness index compared to the other schemes.

Both greedy algorithms applied to 25% of the UEs report identical results in terms of Jain's fairness index and together with the K-means algorithm follow a similar trend as in Figure 6.1a with the decrease of the analysed metric between $N_{FlyBS} = 1$ and $N_{FlyBS} = 3$ and the increase for $N_{FlyBS} > 3$. This anomaly correlates with the explanation provided in Section 6.1. Deploying a small number of FlyBSs, such as in this case 2 or 3, leads to the overloading of the FlyBSs. As the FlyBSs are overloaded, the UEs are allocated with insufficient power from the FlyBS which results in low capacities that are provided to the UEs that are associated to the FlyBSs. On the other hand, the UEs associated directly with the BS benefit from this situation and are provided with significantly greater channel capacities. This results in low values of Jain's fairness index. However, when more FlyBSs are deployed, the number of UEs associated with each FlyBS decreases causing less overloading of the FlyBSs. Therefore, the differences between the channel capacities in the reference cell are reduced which is also followed by the increase of Jain's fairness index. This anomaly does not affect the greedy algorithm applied to all the UEs as this algorithm tends to associate the majority of the UEs to the BS.

As displayed in Figure 6.2a, the proposed final framework provides significantly more fair distribution of the channel capacities to the UEs than the K-means algorithm. This means that K-means algorithm is outperformed by the proposed final framework in all FlyBS counts. The proposed final framework outperforms the K-means algorithm by 11% - 106%. Both the system of the separate DNNs and all the greedy algorithms outperform the final framework in the case when $N_{FlyBS} = 1$, however when the number of FlyBSs stays between $N_{FlyBS} = 2$ and $N_{FlyBS} = 5$, the system of the separate DNNs, the greedy algorithm applied to all the UEs and the greedy algorithms applied to 25% of the UEs deliver worse results in terms Jain's fairness index than the proposed final framework by



(a) N_{FlyBS} dependence, $N_{UE} = 75$

(b) N_{UE} dependence, $N_{FlyBS} = 5$

Figure 6.2: Jain's fairness index depending on the number of FlyBSs and UEs.

2% - 5%, 1% - 6% and 7% - 24% respectively. For cases when $N_{FlyBS} > 5$, the proposed final algorithm, all the greedy algorithms and the system of the separate DNNs converge and deliver very similar values of Jain's fairness index.

As shown in Figure 6.2b, that demonstrates the dependence of Jain's fairness index on the number of the UEs, the values of Jain's fairness index slightly increase for the final framework with the increased number of the UEs in the reference cell until $N_{UE} \leq 50$. After that, a gradual decrease in the value of Jain's fairness value is observed. A very similar trend is also observed for the system of the separated DNNs. Unlike that, all the greedy algortihms report a decreasing trend for Jain's fairness index with the increased number of the UEs throughout all simulation scenarios, with the greedy algorithm applied to all the UEs reporting the best overall performance results in terms of Jain's fairness index. The decreasing trend is also reported by the K-means algorithm that delivers the worst results in terms of Jain's fairness index compared to the other competitive schemes. The decreasing trend with the increased number of the UEs corresponds with the explanation of the decreasing trend of the overall channel capacity provided in Section 6.1 as the general differences in capacities between the UEs associated to the FlyBSs and the UEs associated directly to the BS may be significant.

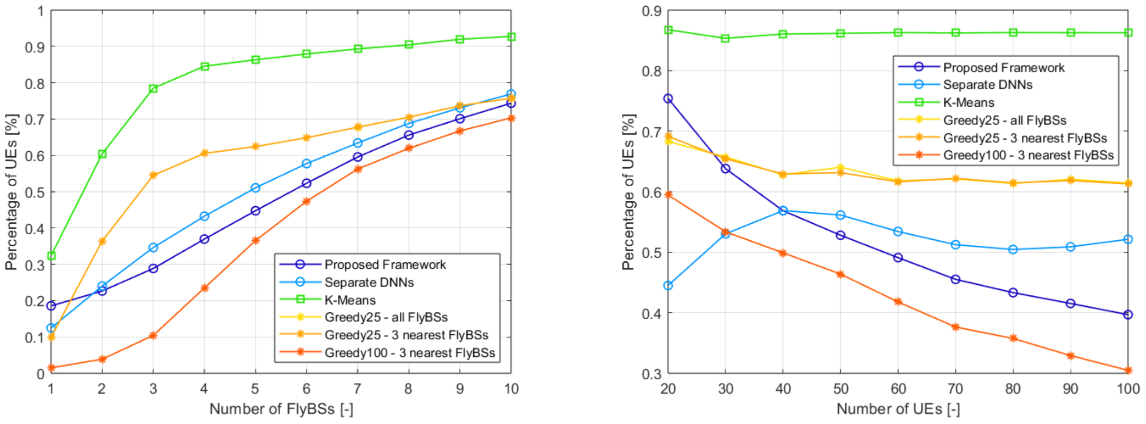
Figure 6.2b demostrates that the final proposed framework provides more fair UE channel capacity distribution than the K-means algorithm by up to 64%. For $N_{UE} \leq 40$, the final framework is outperformed by the greedy algorithms as well as the system of the separated DNNs. For $N_{UE} > 40$, the greedy algorithms applied to 25% of the UEs, which again present almost identical results throughout the simulations, are outperformed by the final framework by up to 8%. Finally, the overall best performance of Jain's fairness index is reported by the greedy algorithm applied to all the UEs in cases when the number of the UEs varies in the simulations.

6.3 UE-FlyBS Association Rate

To analyse how the proposed framework and the other competitive schemes approach the UE association, the rate between the UEs connected via the relay links and the UEs connected to the network directly is inspected in this section. Figure 6.3 displays the percentage of the UEs that are associated to the FlyBSs, thus connected to the network via the relay links.

Figure 6.3a shows how the UE-FlyBS association rate differs with the varying number of the FlyBSs. It could be said that when the number of the FlyBSs in the reference cell increases, a greater percentage of the UEs is associated to the FlyBSs. For the K-means algorithm and the greedy algorithms applied to 25% of the UEs, the percentage of the UEs associated to the FlyBSs rapidly grows until $N_{FlyBS} = 3$. With more FlyBSs deployed in the reference cell, both the K-means algorithm and the greedy algorithm report only a steady growth of approximately 2% per added FlyBS. Both greedy algorithms applied to 25% of the UEs again report identical values as the UEs do not tend to associate to the distant FlyBSs. It is also observed that the K-means algorithm establishes significantly more relay links, in other words more UE-FlyBS associations, in comparison with the other competitive schemes. The final framework and the system of the separated DNNs follow the same trend in the results steadily increasing the percentage of the UEs associated with the FlyBSs. For lower numbers of the FlyBSs in the reference cell, the greedy algorithms applied to 25% of the UEs associates significantly more UEs to the FlyBSs than the final framework. However, as the number of FlyBSs increases, the percentage of the UEs associated to the FlyBSs converge to similar values for the final framework, the greedy algorithms applied to 25% of the UEs and also for the system of the separated DNNs. Finally, the greedy algoritm applied to all the UEs associate the lowest number of the UEs to the FlyBSs in comparison with the other competitive schemes.

As observed in Figure 6.3b, which shows the UE-FlyBS association rate dependence on the number of the UEs, the percentage of the UEs associated to the FlyBS gradually decreases with the increased number of the UEs in the reference cell for the final framework. This may be caused by the DNNs preventing the FlyBSs from overloading. On the other hand, the K-means algorithm and the greedy algorithms applied to 25% of the UEs maintain the percentage of the UEs associated to the FlyBSs at approximately 86% and 62% respectively. The system of the separated DNNs increases the UE-FlyBS association rate with the increased number of the UEs until $N_{UE} = 40$ and then a gradual decrease is observed until $N_{UE} = 80$. This is followed by an increase of the UE-FlyBS association rate for cases $N_{UE} \geq 90$. As observed, the greedy algortihm applied to all the UEs associate the lowest number of the UEs to the FlyBSs for $N_{UE} \geq 40$.



(a) N_{FlyBS} dependence, $N_{UE} = 75$ (b) N_{UE} dependence, $N_{FlyBS} = 5$

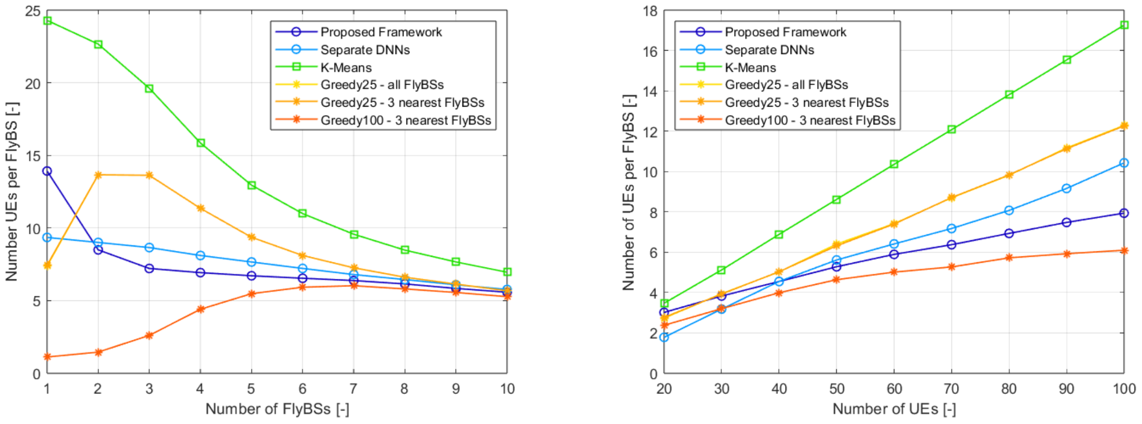
Figure 6.3: UE-FlyBS association rate depending on the number of FlyBSs and UEs.

Considering the overall channel capacity results in Section 6.1 and the percentage of the UE associated to the FlyBSs discussed in this section, it is deduced that the K-means algorithm overloads the FlyBSs with too many UE associations, which leads to worse overall channel capacities and therefore to worse overall performance.

6.4 Average Number of UEs per FlyBS

Finally, the average number of the UEs per FlyBS is discussed. In the similar manner as the metric of the rate of the UEs associated to the FlyBSs in Section 6.3, this metric analyses the UE association policy of the final framework and the competitive schemes. Figure 6.4 shows the dependency of the average number per FlyBS on the number of the FlyBSs and the number of the UEs in the reference cell.

Figure 6.4a shows the results of the average number of the UEs per FlyBS with the varying number of the FlyBSs. The results correspond to the results presented and described in Section 6.3. The average number of the UEs per FlyBS is highest for the K-means algorithm and decreases with the increasing number of the FlyBSs in the reference cell. The final framework keeps lower values of the number of the UEs per FlyBS for all the simulation scenarios in comparison with the system of the separated DNNs and the greedy algorithms applied to 25% of the UEs with the exception for $N_{FlyBS} = 1$ where it reports higher values than the system of the separated DNNs and the greedy algorithm. The greedy algorithms applied to 25% of the UEs associate the lowest number of the UEs per FlyBS for $N_{FlyBS} = 1$, but for $N_{FlyBS} = 2$ and $N_{FlyBS} = 3$ the number of associated UEs per FlyBS rapidly increases. However, in cases $N_{FlyBS} > 3$, the number of the associated UEs per FlyBS reports a steady decrease and converges to similar values as the final framework for $N_{FlyBS} > 8$. Finally, the greedy algorithm applied to all the UEs



(a) N_{FlyBS} dependence, $N_{UE} = 75$ (b) N_{UE} dependence, $N_{FlyBS} = 5$

Figure 6.4: Average number of UEs per FlyBS depending on the number of FlyBSs and UEs.

associates the lowest number of the UEs to the FlyBSs, however the association rate of this algorithm converges to similar values as the final framework for the higher numbers of the FlyBSs.

In the similar way as deduced in the previous section about the percentage of the UEs associated to the FlyBSs, the K-means algorithm establishes more UE-FlyBS associations overloading the FlyBSs that are then unable to provide sufficient channel capacities for the associated UE which leads to the poor performance of in terms of the overall channel capacity in comparison with the other competitive schemes.

In Figure 6.4b the dependence of the average number of the UEs per FlyBS on the number of the UEs in the reference cell is observed. As expected, the average number of the UEs per FlyBS on the number of the UEs increases with the increased number of the UEs. The K-means algorithm reports the highest number of the UEs per FlyBS for all simulation scenarios. This is followed by the greedy algorithms applied to 25% of the UEs, the system of the separated DNNs, the final framework and the greedy algorithm applied to all the UEs. The results correspond to the results shown in Section 6.3.

Chapter 7

Conclusion

The objective of this thesis was to propose a solution that maximizes the overall channel capacity in the mobile network cell where the concept of utilizing the FlyBSs is deployed. The aim of the proposed solution was to optimize the positions of the FlyBSs and the UE associations in order to satisfy the objective of the thesis. For this purpose, a novel framework incorporating the system of the interconnected DNNs was proposed.

The solution is based on addressing the FlyBS positioning and the UE association subproblems in separate ways before combining the solutions of both subproblems into the final framework. The FlyBS positioning subproblem is analysed and then addressed by the system of DNNs whose aim is to predict and then evaluate the shift vectors in whose directions the FlyBSs are repositioned in order to increase the overall channel capacity in the reference cell. The UE association subproblem is also addressed by the DNN that aims to optimize the UE association scheme in the reference cell by predicting the best UE associations for the determined UEs in order to increase the overall channel capacity and therefore increase the overall performance of the network. Later, the solutions of both the positioning and association subproblems are combined into the final framework in the way that the positioning DNNs and the association DNN are interconnected and share information in order to enhance their prediction accuracies.

The performance of the proposed solution has been evaluated and compared with other competitive schemes based on multiple metrics. When evaluating the overall channel capacity, which is the principal performance metric defined in the thesis, the proposed framework outperforms the competitive schemes by up to 14% - 95% depending on the simulation scenarios. Finally, when evaluating the fairness of the channel capacity distribution based on Jain's fairness index, the proposed solution outperforms the competitive schemes by up to 5% - 106%.

The proposed framework considers only static UEs. The future research could explore the complexity of considering UEs that do change their positions throughout the simulations. Another possible direction of enhancing the outcomes of this thesis is to study benefits of applying principles of unsupervised learning algorithms to the UE association as the unsupervised learning methods, such as reinforcement learning, are a promising way how to find optimized solutions for comprehensively addressing the UE association without a necessity to apply the DNN association prediction decision to each UE separately. Finally, future research could be focused on more profound interconnection and interaction between the deployed DNNs as the DNNs would be then potentially capable of identifying more hidden relations in the processed data and, therefore, providing more accurate predictions enhancing the overall system performance.

Appendix A

Attachments

For this thesis, MATLAB R2022a and Jupyter Notebook 7.0.5. were used. The MATLAB M-files, the Jupyter ipynb files and the DNN models (stored as h5 files) developed for the purposes of this thesis are attached. The attached files and models are divided into folders based on their main purposes and functions The folders are sorted in a logical order that follows the steps that were consecutively taken in order to fulfill the main objectives of the thesis. The contents of the folders are summarized in the subsections below.

All the attached files are executable in the specified environments (MATLAB R2022a and Jupyter Notebook 7.0.5.). The codes in the attached files are commented in order to provide a sufficient support when executing the files. Note that for the execution of the majority of the MATLAB files, the MATLAB Parallel Computing Toolbox [22] and MATLAB Deep Learning Toolbox [23] are required.

1. Training Datasets Generation

As the first step, it is necessary to generate the datasets that are later used for the training of the DNNs models. This folder, therefore, contains the MATLAB files used for running the simulations in order to generate the training datasets (saved as csv files). As 3 different DNNs are deployed in the final framework, the folder contains 3 separate M-files, each used to generate the training dataset for a different DNN.

The folder also contains 2 subfolders - *Classes* and *Functions*. The M-files that are in *Classes* folder define the network object types and the system model. The M-files that are in *Functions* folder define blocks of code that are used repetitively in the code execution and therefore it is reasonable for the code transparency to save them as MATLAB functions.

2. Dataset Preparations & DNN Training

When the training datasets are generated, it is time to prepare the datasets for the DNN training and execute the DNN training procedures. Therefore, this folder contains ipynb files runnable in Jupyter Notebook used for the preparations of the datasets for the DNN model training and ipynb files used for the DNN training procedures. The mentioned files are divided according to their functions into subfolders *DNN dataset preparation files* and *DNN training files*. Each subfolder contains 3 files, each corresponding to one of the DNNs.

The folder also contains examples of the training datasets generated in the previous step used for the DNN training that are stored in the folder *Training datasets*. The example datasets each contain 10000 training samples which is generally not a sufficient number of training samples for a proper DNN training. However, as the example datasets serve only for demonstration purposes, the number of samples is considered sufficient.

3. Final Framework DNN Models

This folder contains 3 DNN models deployed in the final framework that were trained by executing the ipynb files earlier. Also, as the models require data scaling, this folder contains corresponding data scaling parameters saved as txt files for each DNN model.

4. Separated DNNs Models

This folder contains 2 DNN models deployed in the competitive scheme System of the separated DNNs that were trained by executing the ipynb files earlier. Also, as the models require data scaling, this folder contains corresponding data scaling parameters saved as txt files for each DNN model.

5. Performance Evaluation

Finally, when the DNNs are trained, the performance evaluation is held. Thus, this folder contains the MATLAB files used for running the performance evaluation simulations. This folder contains 2 M-files for the final proposed framework (one file for the number-of-the-UEs dependence and one file for the number-of-the-FlyBSs dependence), 6 M-files for the competitive schemes (three files for the number-of-the-UEs dependence and three files for the number-of-the-FlyBSs dependence) and 2 M-files for the visualizations of the results. A txt file, that defines the random positions of the UEs for the simulations, is also attached.

The folder also contains 3 subfolders - *Classes*, *Functions* and *Results*. The M-files that are in *Classes* folder define the network object types and the system model. The

M-files that are in *Functions* folder define blocks of code that are used repetitively in the code execution and therefore it is reasonable for the code transparency to save them as Matlab functions. *Results* subfolder contains the numerical values of the simulation results that are to be plotted with the visualization M-files.

References

[1] Y. Zeng, et al., “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Communications Magazine,* , 54(5), pp. 36–42, 2016.

[2] M. Najla, Z. Becvar, P. Mach and D. Gesbert, ”Integrating UAVs as Transparent Relays into Mobile Networks: A Deep Learning Approach,” *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications,* London, UK, 2020, pp. 1-6, doi: 10.1109/PIMRC48278.2020.9217280.

[3] A. Fouda, A. S. Ibrahim, Í. Güvenç and M. Ghosh, ”Interference Management in UAV-Assisted Integrated Access and Backhaul Cellular Networks,” *IEEE Access,* , vol. 7, pp. 104553-104566, 2019, doi: 10.1109/ACCESS.2019.2927176.

[4] A. Madelkhanova, Z. Becvar and T. Spyropoulos, ”Optimization of Cell Individual Offset for Handover of Flying Base Stations and Users,” *IEEE Transactions on Wireless Communications,* , vol. 22, no. 5, pp. 3180-3193, May 2023, doi: 10.1109/TWC.2022.3216342.

[5] J. Plachy and Z. Becvar, ”Energy Efficient Positioning of Flying Base Stations via Coulomb’s law,” *2020 IEEE Globecom Workshops (GC Wkshps, Taipei, Taiwan),* 2020, pp. 1-6, doi: 10.1109/GCWkshps50303.2020.9367495.

[6] M. Najla, Z. Becvar, P. Mach and D. Gesbert, ”Positioning and Association Rules for Transparent Flying Relay Stations,” *IEEE Wireless Communications Letters,* vol. 10, no. 6, pp. 1276-1280, June 2021, doi: 10.1109/LWC.2021.3063909.

[7] G. Amponis, T. Lagkas, M. Zevgara, G. Katsikas, T. Xirofotos, I. Moscholios,P. Sarigiannidis, ”Drones in B5G/6G Networks as Flying Base Stations”,*Drones 2022,* 6, 39, https://doi.org/10.3390/drones6020039

[8] R.D. Alfaia, A.V.d.F Souto, E.H.S Cardoso, j.P.L.d. Araújo, C.R.L. Francês, ”Resource Management in 5G Networks Assisted by UAV Base Stations: Machine Learning for Overloaded Macrocell Prediction Based on Users’ Temporal and Spatial Flow,”,*Drones 2022,* 6, 145. https://doi.org/10.3390/drones6060145

[9] J. Plachy, Z. Becvar, P. Mach, R. Marik and M. Vondra, ”Joint Positioning of Flying Base Stations and Association of Users: Evolutionary-Based Approach,” *in IEEE Access*, vol. 7, pp. 11454-11463, 2019, doi: 10.1109/ACCESS.2019.2892564.

[10] A. Mirzaeinia, M. Mirzaeinia,M. Shekaramiz, M. Hassanalian, ”Placement of UAV-Mounted Mobile Base Station through User Load-Feature K-means Clustering”, 2020, https://doi.org/10.48550/arXiv.2010.01236

[11] Technical Specification Group Radio Access Network, "Study on integrated access and backhaul," *3GPP, Tech. Rep,* 3GPP TR38.874 v16.0.0, Dec. 2018

[12] T. Sap, "Deployment of Flying Base Stations in Emergency Situations,",*Diploma Thesis, Czech Technical University, Faculty of Electrical Engineering, Department of Telecommunication Engineering,* pp. 1-73, 2022.

[13] M. Mozaffari, W. Saad, M. Bennis and M. Debbah, "Optimal Transport Theory for Cell Association in UAV-Enabled Cellular Networks," *IEEE Communications Letters,* vol. 21, no. 9, pp. 2053-2056, Sept. 2017, doi: 10.1109/LCOMM.2017.2710306.

[14] J. Chen, D. Gesbert, "Optimal positioning of flying relays for wireless networks: A LOS map approach, " *2017 IEEE International Conference on Communications (ICC),* 1-6, 2017

[15] M. Najla, Z. Becvar, P. Mach, D. Gesbert, " Predicting Device-to-Device Channels from Cellular Channel Measurements: A Learning Approach," 2019.

[16] P. Mach, Z. Becvar and M. Najla, "Power Allocation, Channel Reuse, and Positioning of Flying Base Stations With Realistic Backhaul," *IEEE Internet of Things Journal,* vol. 9, no. 3, pp. 1790-1805, 1 Feb.1, 2022, doi: 10.1109/JIOT.2021.3088287.

[17] M.T. Hagan, H.B. Demuth, M.H. Bale, O. De Jesús, "Neural Network Design," *2nd Edition,* ISBN 978-0971732117.

[18] A. Lichtner-Bajjaoui, J. Vives, "A Mathematical Introduction to Neural Networks", *Universitat de Barcelona,* 2020, https://diposit.ub.edu/dspace/bitstream/2445/180441/2/tfm_lichtner_bajjaoui_aisha.pdf

[19] S. Damadi, G. Moharrer, M. Cham, "The Backpropagation algorithm for a math student, ", 2023, https://doi.org/10.48550/arXiv.2301.09977.

[20] N. Nouri, J. Abouei, A. R. Sepasian, M. Jaseemuddin, A. Anpalagan and K. N. Plataniotis, "Three-Dimensional Multi-UAV Placement and Resource Allocation for Energy-Efficient IoT Communication," *in IEEE Internet of Things Journal*, vol. 9, no. 3, pp. 2134-2152, 1 Feb.1, 2022, doi: 10.1109/JIOT.2021.3091166.

[21] H. E. Hammouti, D. Hamza, B. Shihada, M. -S. Alouini and J. S. Shamma, "The Optimal and the Greedy: Drone Association and Positioning Schemes for Internet of UAVs," *in IEEE Internet of Things Journal*, vol. 8, no. 18, pp. 14066-14079, 15 Sept.15, 2021, doi: 10.1109/JIOT.2021.3070209.

[22] "Parallel Computing Toolbox," MathWorks. [online]. Available: https://www.mathworks.com/products/parallel-computing.html.

[23] "Deep Learning Toolbox," MathWorks. [online]. Available: https://www.mathworks.com/products/deep-learning.html.