UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**

Planning Self-Backhauled mmWave Networks with Multi-Connectivity

Adriaan P. Dekker

MSc Thesis
Computer Science
Wireless Sensor Systems
August 2021

Supervisors:
prof. dr. J.L. van den Berg
dr. S. Bayhan
Graduation committee:
prof. dr. ir. G.J. Heijenk (chair)
dr. ir. A.B.J Kokkeler (member)

Design and Analysis of
Communication Systems
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

This thesis is the result of the final project that I conducted for my Master degree of Computer Science at the University of Twente. The track of this master program that I had chosen, was Wireless and Sensor Systems. These systems, that can be tiny yet so overwhelming and productive, intrigue me till the current day. This is also the reason why I ended up in the research group Design and Analysis of Communication Systems (DACS), looking for a topic for my final project. Luckily, Hans van den Berg had room to supervise an undergraduate. Since he was involved in a project with a city that was interested in mmWave network planning, this topic came along. Network planning seemed very interesting to me, hence the topic has become my topic, and the thesis has become what it is.

When I got my bachelor certificate after three years of study, I started working as a teacher at a school for secondary education. In retrospect, this was not the best idea, since my master would take another 7 years. Then, there suddenly came a pandemic upon us, which did not make teaching less energy consuming. Luckily, my supervisors were always available, albeit now online. I sincerely hope that the effort that I put in it, will someday save a live or in some other way make the world a better place. The use cases for mere entertainment have never had my heart.

For their indispensable support during this thesis, I would like to thank a number of persons. First of all, I want to sincerely thank Hans van den Berg and Suzan Bayhan for their supervision and seemingly limitless patience. I do not dare to calculate nor guess how many hours I have taken from you. Next, I also would like to thank Geert Heijenk and André Kokkeler for their willingness to take a seat in my graduation committee. Finally, I want to thank my wife, family, and friends for their support, especially in the final weeks before the green-light deadline and the month that followed, while I was finishing my report and preparing my presentation.

Summary

Intelligent Transportation Systems (ITSs) are expected to require large amounts of bandwidth, while aiming to improve road safety, traffic efficiency, and traveling comfort. Beside a large bandwidth, some services also require Ultra Reliable Low Latency Communication (URLLC), especially those concerning safety. MmWave communication is regarded as primary enabler for these purposes, since the abundance of spectrum at those frequencies offers much bandwidth. However, due to bad propagation of mmWave signals, dense network deployment is needed. The proneness to blockages further increases the need for densification, since Multi-Connectivity (MCo) is often used to account for this. To reduce deployment costs, self-backhauling has been introduced to reduce expensive fiber deployment.

In this thesis, we study the Cell Planning (CP) problem for mmWave networks in an urban context. Most previous works on the issue of mmWave CP consider characteristics of such environments, like the high probability of blockages by moving obstacles, and the clustering of users at certain hot spots. Extensive research has been carried out towards mmWave CP with MCo, or with self-backhauling. However, it is not clear how CP should be performed when using both MCo and self-backhauling.

Therefore, in this study we deliver design insights on CP integrating both MCo and self-backhauling. We design a computationally efficient mmWave cell planning algorithm to run simulations. The algorithm uses a greedy approach to deploy BSs, and adds optimization phases. Optimization is done for two objectives. First and foremost, the deployment costs are minimized. Both costs for Base Stations (BSs) and backhaul fibers are considered. Next, without increasing the costs, the BSs are placed such that the chances of multiple simultaneous link blockages are reduced, providing access reliability. We plan the network such that each test point is covered by at least K BSs (K-coverage), of which at least one has a fiber backhaul.

We compare the performance of our algorithm to a greedy baseline algorithm, and in a small scenario to a brute force algorithm. As expected, the optimization steps have the highest time complexity, which is polynomial to the number of test points (TPs) to be served ($O(n_{TP}^3)$. We find that our algorithm is suitable for running simulations. These simulations result in the following design insights. First, an environment with less static blockages enables better access reliability. On the other hand, when many static blockages are present, access reliability is dramatically decreased when the number of links provided to each test point is increased. Next, using self-backhauling with MCo provides flexibility to increase access reliability without increasing the system costs. Additionally, using self-backhauling with MCo does not significantly increase the number of required BSs with a fiber backhaul. Finally, we conclude that using self-backhauling is a cost-effective approach for mmWave CP with multi-connectivity and we provide a list of interesting future research directions.

Contents

List of acronyms

A-BS	anchored base station
AD	angular diversity
AR	augmented reality
AV	autonomous vehicle
BF	beamforming
BH	backhaul
BS	base station
CAV	connected and autonomous vehicle
CA	carrier aggregation
CDSA	control and data plane split architecture
CoMP	coordinated multi-point
CP	cell planning
CS/CB-CoMP	coordinated scheduling/coordinated beamforming coordinated multi-point (CoMP)
CSI	channel state information
CS	candidate site
DC	dual connectivity
DPS-CoMP	dynamic point switching CoMP
FAP	fiber access point
GA	genetic algorithm
GC	greedy construction
HPPP	homogeneous poisson point process
MIND-G	MCo with IAB mmWave Network Deployment using Greedy construction (without optimization steps)

MIND-GO	MCo with IAB mmWave Network Deployment using Greedy construction with Optimization
IAB	integrated access and backhauling
ISI	inter symbol interference
ITS	intelligent transportation systems
JT-CoMP	joint transmission CoMP
LL	link length
LoS	line-of-sight
MC	macro cell
MCo	multi-connectivity
MEC	multi-access edge computing
MIMO	multiple-in multiple-out
mmWave	millimeter wave
NLoS	non-line-of-sight
NSGA	non-dominated sorting genetic algorithm
P-BS	primary base station
PSK	phase shift keying
QAM	quadrature amplitude modulation
QoS	quality-of-service
RSS	received signal strength
SC	small cell
SFN	single frequency network
SINR	signal to interference and noise ratio
SNR	signal to noise ratio
TP	test point
UDN	ultra dense network
UE	user equipment
URLLC	ultra-reliable low latency communication
VR	virtual reality
W-BS	wireless base station

Chapter 1

Introduction

Future smart city applications, e.g. intelligent transportation systems (ITS), are expected to require communication networks with large amounts of bandwidth. Connected and autonomous vehicles (CAVs) will be responsible for a large portion of this expected demand [1], [2]. The main sources of these large data streams are services that improve road safety, traffic efficiency and traveling comfort. Noteworthy examples of exchanged data are (1) sensor data for collective perception, (2) control information for platooning, (3) vehicle trajectories for collision prevention, and (4) mobile broadband internet for multi-media streams [3].

Some of those services do not only require large bandwidth, but also put constraints on the latency of the communication. Especially when safety is concerned, the latency has to be below a certain threshold value with a certain reliability, e.g. 99.999% below 10ms [4]. This is called ultra-reliable low latency communication (URLLC).

Millimeter wave (mmWave) communication is expected to comply with these requirements, since much spectrum is available at mmWave frequencies [5]. The utilization of mmWave radio communication, however, is complicated by its propagation characteristics. Due to high atmospheric absorption and high penetration loss, signals have a very limited range and are prone to blockages [6], resulting in the need for a very dense network deployment. In particular, line-of-sight (LoS) communication is preferred over non-line-of-sight (NLoS) communication, since this influences the path loss exponent drastically [2]. To achieve an acceptable range of around 150 meters, beamforming (BF) is used, but dense deployment is still required [7]. To combat link loss due to blockages by providing a higher LoS link probability, multi-connectivity (MCo) has been introduced, which requires an even denser deployment of base stations (BSs) [8]. This results in the challenge of how all BSs can be provided with an appropriate link towards the core network (backhaul link) in a cost-efficient way, and how this must be incorporated in cell planning (CP) [9].

Wireless backhauling is seen as a good way to reduce the deployment costs of a network. Since there is abundance of spectrum at mmWave frequencies, a wireless backhaul (BH) link can provide nearly the same link quality as a fiber link [7]. When the same spectrum is used for both user equipment (UE) links as for BH links, the terms self-backhauling or integrated access and backhauling (IAB) are used. Moreover, the BSs that are wirelessly backhauled are assumed to be less expensive because no fiber connection hardware is needed [9].

Previous work on mmWave CP focuses mainly on deploying BSs under some quality constraints, while minimizing the costs. Quality constraints that are often used include LoS coverage, average signal to interference and noise ratio (SINR), received signal strength (RSS) threshold, or rate coverage probability, to name a few. These metrics are often assessed for a number of test points (TPs), or for tiles. TPs are locations that represent one or more users, and tiles are portions of the map. The difference between these two approaches is that in the former case only at some locations a TP is present, and in the latter case the entire map is divided into tiles. The deployment costs are mostly measured as the number of placed BSs.

One aspect that is not so much explored in mmWave CP literature, is nomadic blockages [8]. Blockages can be divided into two categories, namely static and dynamic blockages. *Static blockages* are for example buildings. Those are taken into account in many studies, some even using real city maps [1], [7], [10]. The *dynamic blockages*, however, are not so much investigated, although they play a critical role for the performance of mmWave links, especially in crowded areas like cities. See Figure 1.1 for an example scenario in which two simultaneous blockages cause that the signal quality for the user is suddenly very bad. A well-known technique to cope with these dynamic

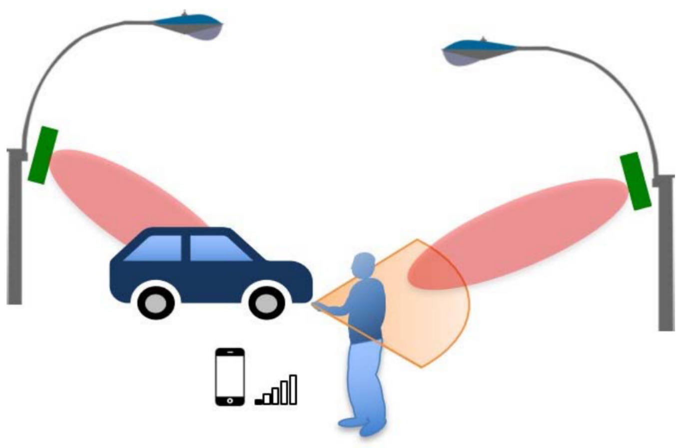

Figure 1.1: Examples of dynamic blockages: a car and one's own body [8]

blockages is MCo, since it provides users with additional links in case the primary link is suddenly blocked. Most of the time when MCo is studied, it is assumed that all BSs are backhauled “for free”. We found that studies only include the costs of fiber backhauling when they aim to investigate self-backhauling too.

To the best of our knowledge, no research has been done on the use of IAB in combination with MCo. Therefore, we formulate the following research questions that will be answered in this thesis:

> *Given a set of candidate sites, at which of those locations should mmWave base stations be placed to provide the required coverage with appropriate quality of service for users in a city road environment while keeping costs as low as possible?*

In this question, the required coverage refers to the number of links that must provide sufficient capacity to each user (MCo). It is also accounted for that the quality of service is only improved by MCo if dynamic blockages cannot block multiple links simultaneously, which is for example possible when the angular separation between two links is very small. To keep the costs as low as possible, IAB will be applied. To investigate the implications of MCo and IAB, the following sub-questions are formulated.

Q1. *What are the implications of the requirement for multi-connectivity on mmWave BS placement?*

Q2. *What are the implications of using integrated access and backhauling on mmWave BS placement?*

These questions are answered by designing a computationally efficient heuristic algorithm for planning a network with both MCo and IAB. This algorithm is named “MCo with IAB mmWave Network Deployment using Greedy construction with Optimization”, shortly MIND-GO. By setting appropriate constraints for the required coverage and then minimizing the costs, the MIND-GO algorithm provides a solution to our main research question. The MIND-GO algorithm also optimizes for blockage robustness, as long as this does not increase the network deployment costs. The sub-questions are answered by adjusting parameter values of our algorithm and measure the performance of the resulting network. An answer to Q1 is given by adjusting the MCo parameter for the number of BSs that should cover each TP. Question Q2 is investigated by comparing network deployments using IAB with those that provide each BS with a wired backhaul.

The remainder of this thesis is organized as follows. Chapter 2 gives an overview of relevant literature for this thesis and presents the work on which we build our contributions. In Chapter 3, the system model is defined. Then, the optimization problem is given in Chapter 4. This is used to define the problem clearly and gain insight to develop a heuristic algorithm. The variables from this optimization problem are also used in that algorithm. Chapter 5 provides an overview of the phases of our algorithm, and details how each phase works and contributes to an effective solution. The performance evaluation of the algorithm is carried out using a novel urban environment simulator that we built in MATLAB. Chapter 6 presents the results of all simulations, in which the efficiency of the algorithm is explored by comparing it to a brute force optimal algorithm and to a greedy non-optimized variant of the MIND-GO algorithm. Furthermore, the implications of using MCo and IAB are analyzed to answer the subquestions. We expect these results will contribute to the introduction of high-quality network deployments that can facilitate smart city applications and CAVs. Finally, Chapter 7 answers our research questions and lists suggestions for future work.

Chapter 2

Background and related work

This chapter provides a background for our study based on existing research towards the use of mmWave channels for intelligent transportation systems (ITS) scenarios. Furthermore, we will elaborate on current research directions in cell planning (CP), especially involving techniques that are relevant for mmWave CP. Finally, we present the contributions of this thesis to the field of mmWave CP.

2.1 Using millimeter wave channels in ITS scenarios

Use cases for mmWave links share one common need: a wireless connection with a high throughput. One example scenario for this is (semi-)autonomous driving because much sensor data needs to be shared and acquired [11]. In this scenario, other streams of data add up to this, such as the provision of autonomous vehicle (AV) passengers with comfort services when travelling [3]. To enable high throughput, mmWave links can generally be used in two ways: firstly, as a connection between user equipments (UEs) and base stations (BSs), and secondly, as means of forwarding data in the edge network (fronthaul) or to the core network (backhaul). The relevance of those use cases for ITS is explained below.

Kong et al. [6] propose the use of V2N mmWave links to determine the optimal driving strategy by sharing data with servers in the network edge utilizing multi-access edge computing (MEC). Two examples they give for this are (1) processing sensor data to recognize objects in real-time, and (2) to fill measurement gaps and observe blind areas. Elbamby et al. [4] too describe mmWave communication as the ultimate solution to send data to MEC, thus enabling computer vision and image processing for ITS scenarios. To assist AVs with such MEC-aided services, sensor data streams of approximately a Gigabit per second have to be gathered and processed with a latency below 10 ms and a reliability of nearly 100% [12].

Those two examples mainly use uplink mmWave channels. Services using large amounts of bandwidth in the downlink, are for example virtual reality (VR) and augmented reality (AR) applications, such as gaming or remote conferencing [4]. The main idea behind them is to supply travellers with an internet connection which is as good as at home or at work, so that they are not limited to be productive or to entertain themselves when on the road [3].

Most of the data that needs to be transmitted from and to the vehicles must be forwarded by the mmWave BSs towards their destination. Some BSs might include a MEC server which can process the data and then return the results to the vehicle, but most BSs will likely need to forward the data to another BS with MEC capabilities or to the core network. This requires a high throughput network which is more flexible and scalable than fiber links [13]. Hence, BSs should be interconnected by fronthaul links to reduce the load of the backbone router. For this purpose, mmWave links can be used, thus eliminating the need for wired connections to be deployed between BSs and realizing a flexible network topology [14].

The remainder of this section describes three aspects of mmWave that are relevant for our study. First, we give an overview on mmWave channel characteristics. Next, we describe the use and challenges of beamforming (BF). Then, the benefits of different forms of multi-connectivity (MCo) are discussed. Finally, the use of integrated access and backhauling (IAB) is outlined.

2.1.1 Millimeter wave channels: applications and challenges

Three aspects of mmWave channels determine how they can be used, namely the available frequency bands, the harsh propagation, and the achievable throughput.

First of all, mmWave channels are all radio links utilizing a frequency above 6 GHz, up to 300 GHz. In Figure 2.1 (copied from [13]), an overview of suitable mmWave frequencies below 90 GHz is given that are available in Korea, the EU and the US. These include the commonly researched frequencies, namely 28 GHz [13], [15], [16] 60 GHz [13], [17], and 73 GHz [13], [18], since much unused and unlicensed spectrum is available at those bands.

Secondly, the main problems of high frequency radio signals are damping and occlusion. Damping is already significantly incurred by the oxygen in the air. For example, the propagation loss of 60 GHz signals is 1.6 dB/100 m, which is increased to 3.6 dB/100 m in heavy rain of 50 mm/h [6]. Solutions for damping are accomplished by using BF and dense deployment of cells, which is elaborated on in Section 2.1.2. Then, the second problem however comes into play, because penetration of solid objects leads to much worse signal degradation, referred to as occlusion. Only the human body will

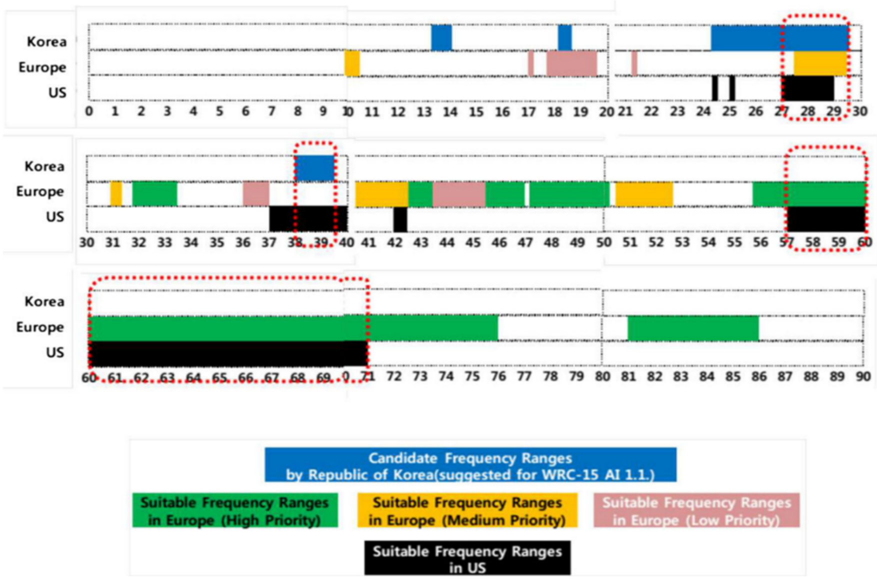

Figure 2.1: Candidate mmWave frequencies in Korea, EU and US (GHz) [13]

already result in signal loss of 20-50 dB, decreasing the communication range to only a couple of meters [6]. Common building materials like brick even completely block the signal [15]. Some materials like metals and glass reflect the signal very well, but this also increases the complexity of determining the optimal beam direction since good beam directions might now be completely unrelated [19]. Although NLoS paths can be used, this is observed to increase the path loss exponent from under 2.8 to over 3.8 [2]. Therefore, primarily LoS paths should be used to communicate over mmWave channels, making it very difficult to use in dynamic and obstacle-rich environments like city traffic. A solution for blockage is expected from using MCo, as discussed in more detail in Section 2.1.3.

Finally, the achievable throughputs with mmWave links are expected to be multiple Gigabits per second (Gbps) [11], [12], [20], [21]. This is mainly enabled by the availability of much bandwidth in the mmWave spectrum. In the D-band only (110 GHz - 170 GHz), 40 GHz of suitable bandwidth for gigabit communication is available [22]. With spectral efficiency just 1 bps/Hz, a total throughput of 40 Gbps can already be realized. Higher spectral efficiency is enabled by BF [13] and the use of modulation techniques like multilevel phase shift keying (PSK) or multilevel quadrature amplitude modulation (QAM) [22]. For example, 802.11ad uses 2.16 GHz of spectrum in the 60 GHz band, enabling throughputs up to 7 Gbps [5] at a range of 5 to 10 meter [23], reaching a spectral efficiency of up to 3.24 bps/Hz. In another research, throughputs of up to 450 Mbps per user have been reached with only 100 MHz of bandwidth in the 28 GHz band with a cell radius of 100 m and antenna gain of 20 dB in a 30° wide beam [24]. This indicates that with similar equipment and a bandwidth of 2 GHz, throughputs of 9 Gbps could be reached.

2.1.2 Beamforming for millimeter wave frequencies

Beamforming (BF) is the primary technique to deal with the bad propagation characteristics of mmWave links [11]. In this section, two aspects of BF are described. Firstly, some aspects of beam forming antennas are surveyed to give a sense of the possibilities and constraints of BF for mmWave communication. Secondly, two important control tasks for using BF are summarized.

Millimeter-wave beamforming antennas

The BF technique utilizes multiple antenna elements to concentrate the radio signal in one direction. The number of antenna elements used in a system determines the width and length of the beam: more elements result in narrower beams with a larger reach. For example, with an 4×4 multiple-in multiple-out (MIMO) antenna array, beams of 45°can be formed, whereas an 8×8 MIMO antenna array enables beams of 22,5° [15]. Beside the main lobe, some side lobes are always formed. More antennas also result in more and smaller side lobes as well as a stronger and narrower main lobe. Having more antenna elements therefore results in better mmWave signal quality. Luckily, the short wavelength of mmWave frequencies enables to incorporate many antenna elements in small devices [4], [13]. Therefore, UEs will also be able to use BF.

BF directs most of the effective isotropic radiated power in one direction, called the main beam or main lobe. Only a fraction of the power is radiated in the so-called side lobes. In many studies, these are not even considered, to simplify the system model. Used antenna gains at 28 GHz vary from 24.5 dBi with a beam width of around 10° in [25], [26] to 10-15 dBi for beam widths of around 30° in [26], [27]. Clearly, there is a trade-off between antenna gain and beam width. The best beam width depends on the scenario, since higher beam widths allow for higher mobility [28].

A receiver can also apply BF to listen in a particular direction. This decreases the effect of interference, since signals from other directions are filtered out. On the other hand, control tasks are further complicated, since both beams have to be aligned to achieve optimal channel quality. Some resulting challenges are discussed below.

Beam management

The alignment of beams between sender and receiver involves a set of control tasks, which are referred to in literature as beam management [29]. Two challenges on this subject are beam selection at initial access and beam tracking to keep beams aligned. Both are explained below.

When a new UE enters a mmWave network, a beam direction has to be determined that results in a sufficient signal strength. At sub-6 GHz frequencies, this can be done by sending a signal in all directions and then determining the direction of the UE response. With mmWave links, this is a greater challenge, since propagation is harsh at these frequencies, hugely limiting the reach of a broadcast signal. Therefore, solutions must be exploited that do not rely on broadcast signals. Three examples are depicted in Figure 2.2 (copied from [20]) and described below, based on the work of Giordani et al. [20]. First, in “exhaustive search”, both the UE and the BS can sweep the surrounding space to find optimal sending and listening beam directions. This strategy consumes many time slots, because all beam direction combinations must be checked, but it results in an optimal pair of beams. Second, an iterative approach can be used, which first searches with wider beams and then determines which narrower beam in that direction is optimal. This delays the initial access procedure less, but it also limits the reach in which a UE can be detected. Lastly, a third solution is to use other technologies to determine the location of the UE relative to the BS and communicate this via a sub-6GHz channel. For this, multiple solutions beside GPS are proposed, including using triangulation with mmWave signals [16] or other RAT signals [30].

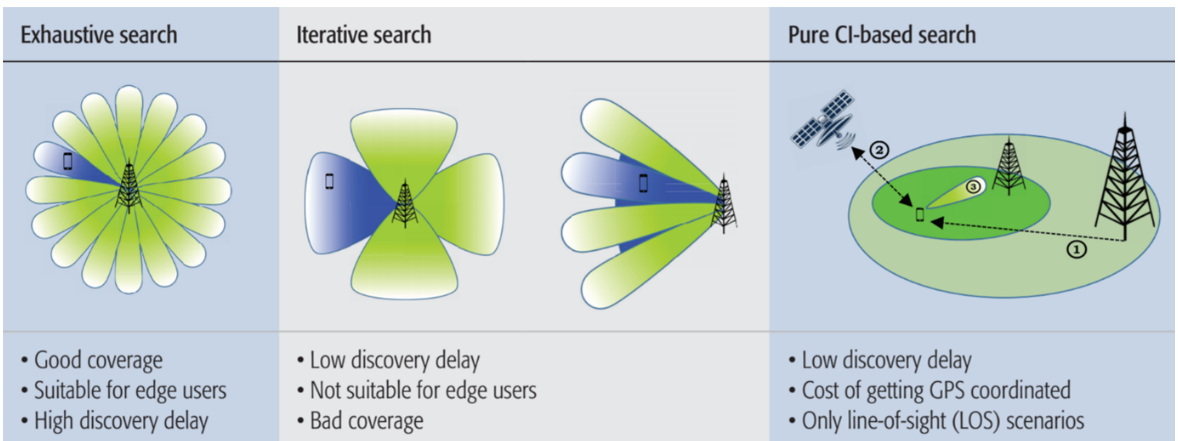

Figure 2.2: Summary of three initial access strategies [20]

When using BF, links can be interrupted because the beam direction of the sender and the listening beam direction of the receiver are misaligned. This phenomenon is also referred to as deafness [20]. Clearly, the direction at which the beam should be aimed has to be updated when a UE has moved. This is called beam tracking [4]. Each time beam tracking is performed, the BF vector is updated to steer the beam in the new direction of the user. Especially when UEs are moving fast, this is extra challenging, since then the movement of users must be precisely tracked to prevent deafness. Using smaller beam widths could help too, but this decreases the signal reach and increases the interference. Hence, the authors of [31] propose a beam tracking scheme in which the update interval is dependent at the UE's velocity. This also takes some bandwidth, which is assumed to be 20% of total bandwidth [24]. Because the LoS path might suffer from blockage or damping, well-reflecting surfaces might deliver better NLoS paths [19]. This further complicates the process of adapting the beam direction to the new UE position. To find optimal paths towards UEs, machine learning is utilized [11], [19].

2.1.3 Multi-connectivity for millimeter wave access reliability

The main goals of multi-connectivity (MCo) are to enhance throughput and reliability of radio links [4]. In this section, the general use of MCo will be described and then some examples will be given on how it can be applied to mmWave networks.

General types of MCo

Four types of MCo are distinguished in [4], of which two are classified as inter-frequency and two as intra-frequency. The different types of MCo are schematized in Figure 2.3 (copied from [32]).

Intra-frequency MCo is applied in coordinated multi-point (CoMP) communication and in single frequency networks (SFNs). In CoMP, BSs are operated as a distributed multiple antenna system. Therefore, strict coordination is needed to achieve a gain in signal strength and to prevent interference. In SFNs, data is transmitted by multiple antennas on the same frequency and at the same time. If their propagation delays towards the receiver are tightly bounded, the message can be reconstructed at the receiver, else this will result in inter symbol interference (ISI).

Examples of inter-frequency MCo solutions considered most noteworthy in [4], are carrier aggregation (CA) and dual connectivity (DC). CA is a technique in which multiple frequency blocks are simultaneously used by the same UE and the scheduling and interference management of these are orchestrated to achieve a higher bandwidth. DC utilizes a combination of different network technologies to enhance communication robustness, e.g. by using one of the links as fallback if the other link fails.

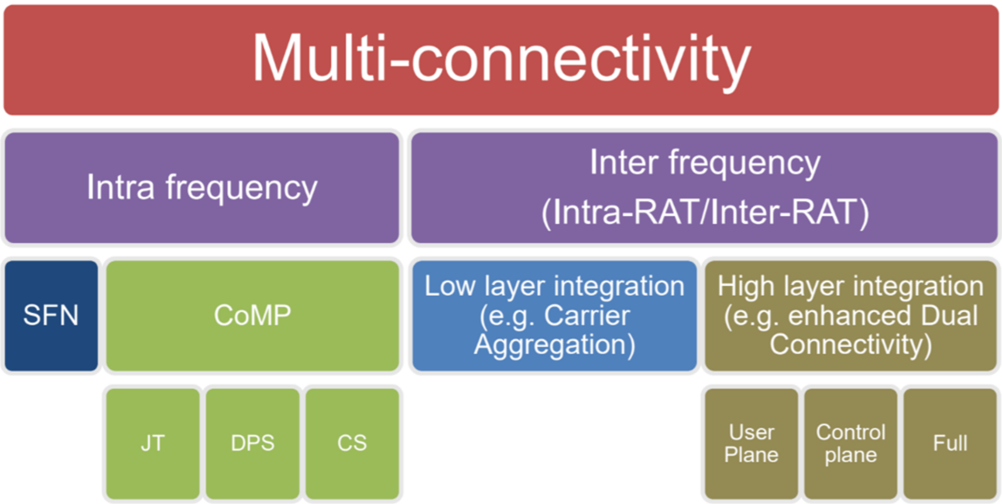

Figure 2.3: Different types of multi-connectivity [32]

Three of the four given examples of MCo are regularly applied with mmWave: CoMP, CA, and DC. The CA technique is applied as described above, so this will not be further explained. CoMP and DC, however, do serve very specific purposes, hence they will be elaborated below.

Coordinated multi-point with mmWave

CoMP is used to counteract the problems of blockage and deafness. Three forms of this technique are characterized by the mmMAGIC project [32], namely joint transmission CoMP (JT-CoMP), dynamic point switching CoMP (DPS-CoMP), and coordinated scheduling/coordinated beamforming CoMP (CS/CB-CoMP). Figure 2.4 (copied from [32]) depicts how user plane messages are routed through the system in the first two CoMP variants. All named CoMP variants use some form of coordination between BSs to enhance the channel quality, requiring strict synchronization between those nodes. Of these, JT-CoMP is not suitable for use with mmWave links, since this requires ideal front hauls [32]. Furthermore, the UE must be able to listen in multiple directions at the same time. However, this is impossible when analog BF is used at either the BS or UE side, which is very likely for the UE [15].

Coordination between BSs can nonetheless be very useful for mmWave beams to enable fast switching between data paths with DPS-CoMP in case of radio link failure [15] or to improve beamforming accuracy with CS/CB-CoMP. In the former case, multiple connections are available to a UE simultaneously. In the latter case, BSs only share channel state information (CSI) data, so UEs are not actively involved in realizing this type of CoMP. The use of DPS-CoMP has been proposed as a particularly suitable solution to deal with the high blockage probability of mmWave links [8]. This means

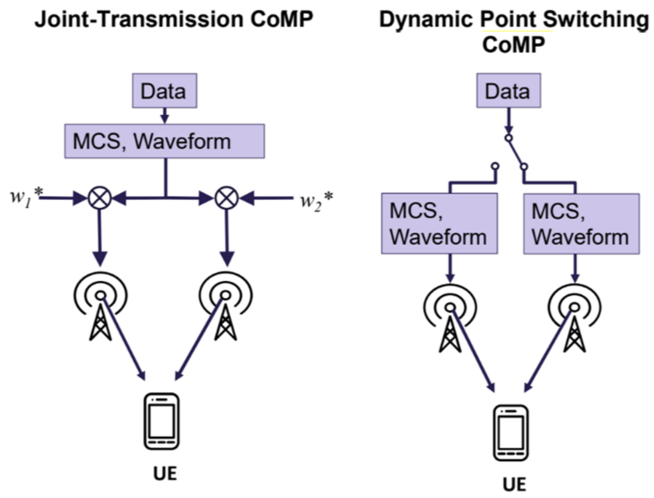

Figure 2.4: Two examples of CoMP solutions [32]

that a UE has multiple connections to BSs simultaneously, one of which is the active (primary) connection. The other connections are secondary links, to which communication can be switched when the primary link is blocked or if a proactive handover is necessary due to user movement.

Dual connectivity with mmWave and sub-6 GHz

In [33], it is proposed to use DC with mmWave and microwave (sub-6 GHz) links to support the bandwidth offered by mmWave links with a highly reliable microwave connection. An example of this setup is depicted in Figure 2.5 (copied from [33]). This is useful to keep a UE connected to the network, even when a mmWave link suddenly fails [4]. Because a microwave connection is much more robust than mmWave links, Giordani et al. [15] propose to use this as fallback for control signalling when handover and scheduling decisions cannot be communicated over established mmWave links. According to the 5G PPP Architecture Working Group [34], control and data plane split architecture (CDSA) should be used to provide a unified control framework for 5G. When considering mmWave small cells, the benefits of this approach are clear. User plane data can be transmitted at high rates using mmWave, while control messages are exchanged over much more robust channels of sub-6 GHz macro cells.

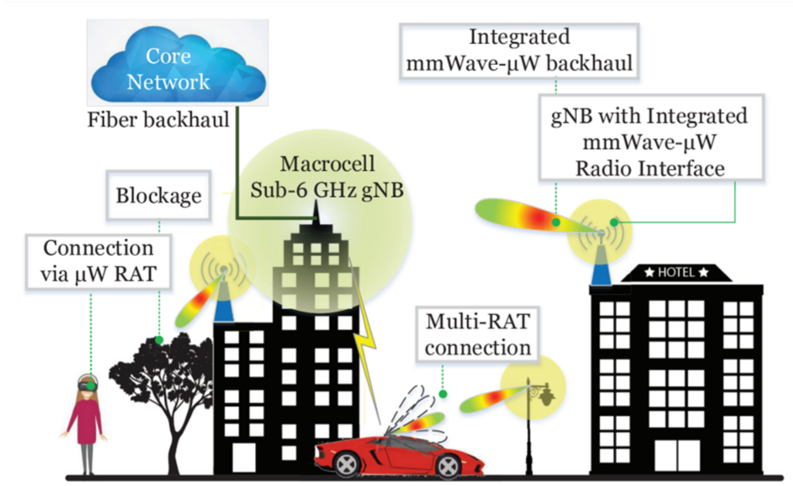

Figure 2.5: Example of intra-frequency MCo with sub-6 GHz and mmWave BSs [33]

2.1.4 Millimeter wave with self-backhauling

Self backhauling is a technique in which the same frequency band is used for both access and backhaul links. Earlier forms (like LTE relaying) were not successful due to high interference, resulting in significant loss of throughput and increase in latency [7]. This is not expected to be a problem with mmWave, because the use of narrow beams results in interference isolation. However, the use of multiple hops still introduces a lower throughput and a higher latency [35], resulting in a 1-hop maximum for some use cases [32].

The main reason to apply self-backhauling is that it provides a cost-effective way of backhauling ultra dense networks [7], [36]–[38]. Another positive effect is the greater flexibility for the deployment of networks, enabling them to be rolled out in stages [32]. This also enables a smooth transition to 5G mmWave networks [39]. Polese et al. [35] show that densification using IAB performs almost as well as an all-wired setup, when UEs are mainly watching videos and visiting web pages. If the network is saturated with traffic, this does not hold.

Saadat et al. [36] identify resource allocation as a major concern. The radio resources can be shared among access and backhaul links by either multiplexing in frequency or time, or by reserving two different sets of bandwidth resources. The former is commonly referred to as integrated access and backhauling (IAB), although in many cases the terms self-backhauling and IAB are used interchangeably. In [40], it is shown that IAB results in a significantly more complex resource allocation scheme compared to orthogonal resource division. On the other hand, it is also the most flexible option of resource allocation. As a third option, separate antennas can be used for access and backhaul links in combination with self-interference cancellation techniques [9]. This way, an IAB-node can transmit and receive data simultaneously, at the cost of more complex and hence also more expensive equipment.

2.2 Millimeter wave cell planning

In general, the cell planning (CP) process consists of three phases: dimensioning, detailed planning, and optimization, as shown in Figure 2.6 [41]. Even the simplest version of cell planning is known to be an NP-hard problem, since it is an extension of the minimal cost set covering problem [42]. The solution space is huge, even for relatively small cellular systems. The advent of mmWave does complicate this problem even further [37]. So, using brute force to find an optimal solution is absolutely undoable. Therefore, much research is done on deployment algorithms that approach an optimal solution. Such heuristic algorithms try to reduce the complexity of the problem, to find a good approximation within reasonable computation time [41].

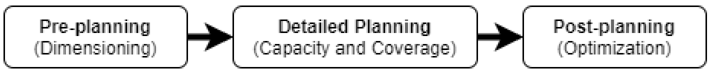

Figure 2.6: Three phases of CP, based on [41]

CP algorithms can be developed for different goals. Some algorithms are designed for detailed planning of small or medium sized network deployments. Such algorithms often use an evolutionary approach, of which the genetic algorithm (GA) is the most well-known. For example, Athanasiadou et al. [43] provide a generic analysis of ultra dense network (UDN) deployment with mmWave, for which they minimize (1) the deployment costs, (2) the number of users not meeting their capacity and coverage constraints, and (3) the excess resource blocks needed at network level. They construct a Pareto front for their optimization problem using the non-dominated sorting genetic algorithm (NSGA) [43], which takes a solution set and evolves this by selecting the best solutions from the set and then create new solutions based on this best 'population'. This approach is notoriously computationally expensive but generates high quality solutions [1].

When larger scenarios are considered or extensive simulations have to be run, approximation-based algorithms with a much better computation efficiency are developed. Many of these are greedy construction (GC) algorithms. Such an algorithm is developed for example by Wu et al. [44] to provide an time-efficient algorithm for mmWave CP. This algorithm is aimed at minimizing the network deployment costs under quality-of-service (QoS) constraints like the rate coverage probability.

Below, two specific scenarios of mmWave CP are presented. These deal with the aspects that are considered in this thesis, namely multi-connectivity and self-backhauling.

mmWave CP with MCo

For data-link mmWave communication, DPS-CoMP can be used to perform fast handovers in case of blockages. To promote the chance for a suitable alternative link, BSs must be close to users and preferably located in different directions. In [8], Devoti et al. found that this can be effectively scored by a combination of link length (LL) and angular diversity (AD), in which LL is determined as the average link length and AD as the average of the minimal angles between any pair of links for each UE. This is the first work to propose and test this approach. We could also not find any other work incorporating MCo in mmWave CP.

CP with mmWave backhauls

Rezaabad et al. [9] propose an NSGA algorithm for planning a 5G mmWave small cell (SC) network, either with or without self-backhauling. For wireless backhaul links, separated highly directional horn antennas are assumed to limit interference with user access links. In their model, they include costs for deploying a cable between a BS to a fiber access point (FAP), for the installation of an optical splitter at the FAP, and for using the feeder fiber. Using the algorithm, the costs of BS deployment are minimized and the number of users whose throughput requirement is satisfied is maximized. Constraints are used to ensure cell coverage and provide sufficient capacity in both access and backhaul links.

Muñoz et al. [37] propose a heuristics based algorithm for planning additional mmWave small cells with mmWave backhauls in existing HetNets with macro cells (MCs) and SCs. MCs are provided with a fiber backhaul, and all SCs are connected via a mmWave backhaul link to exactly one other SC or an MC. Two steps are taken in the algorithm: (1) provide coverage for user demand and (2) adapt to satisfy backhaul constraints. For this, blockages are not considered, neither is MCo. An SC within LoS of another MC or SC is considered connected with the core network.

We conclude that not much research has been carried out toward mmWave CP with MCo. More research has been done on the field of mmWave backhauls and, more specifically, mmWave IAB, but the influence of this on the CP process is not fully understood yet. For example, the use of IAB sounds very promising in combination with MCo, but we also found no studies on that subject. Therefore, our contributions will be on this subject, as presented in the next section.

2.3 Contributions

For this thesis, a computationally efficient CP algorithm will be developed. The goal of this is two-fold: Firstly, the algorithm is developed to provide design insights in the deployment of BSs when using both MCo and IAB. Secondly, it can be used as a tool for the first phase of the actual CP process, namely pre-planning, and it may be expanded to serve for the detailed planning as well. The relevant research contributions are listed as follows.

- **Novel approximation algorithm**
  Like the algorithm in [1], we design an algorithm that consists of multiple phases, of which two phases are greedy. Along the greedy phases, three optimization phases are defined. However, unlike the algorithm of [1] that minimizes the cost under QoS constraints, we will minimize the costs and then, as long as it does not increase the system costs, maximize the access reliability simultaneously.
- **Deploying mmWave BSs with MCo**
  To counteract the blockage proneness of mmWave links, our algorithm will be able to plan a K-coverage network for any K. Of course, the deployment may fail if not enough candidate sites (CSs) are provided to cover all TPs sufficiently.
- **Deploying mmWave BSs with IAB**
  In addition to the deployment for access reliable MCo as in [8], we will also include IAB to reduce the network deployment costs. We will consider the costs of using fiber backhauls in the same way as done in [9].
- **Apply an access reliability metric**
  In our algorithm, access reliability will be measured as the angular diversity and link length score, as presented in [8]. This means that we will try to minimize the average link length, and to maximize the average angular diversity. Angular diversity is determined by the smallest angle between two links that connect a TP to a BS, considering all links of a TP.

In short, this thesis contributes to cell planning theory by designing a planning algorithm for the deployment of a mmWave small cell network, considering both multi-connectivity and self-backhauling. To the best of our knowledge, no such research has been performed earlier. We specifically consider ITS scenarios in which vehicles and their passengers are provided with a broadband internet connection.

Chapter 3

System model

In this chapter, we explain how we modeled our system for network deployment. The system consists of three main components: (1) *test points (TPs)* representing the user demand; (2) *fiber access points (FAPs)* providing internet access over fiber cables; and (3) *candidate sites (CSs)* at which BSs can be deployed. A BS can either be anchored (anchored base stations (A-BSs)), meaning it is connected by fiber to an FAP, or wirelessly backhauled (wireless base stations (W-BSs)) via an A-BS. It must be noted that all A-BSs together must have sufficient capacity to provide all TPs for their needs, since their data has to be sent by an A-BS, either directly or via a W-BS. Figure 3.1 gives a schematic overview of the possible connections between the system components.

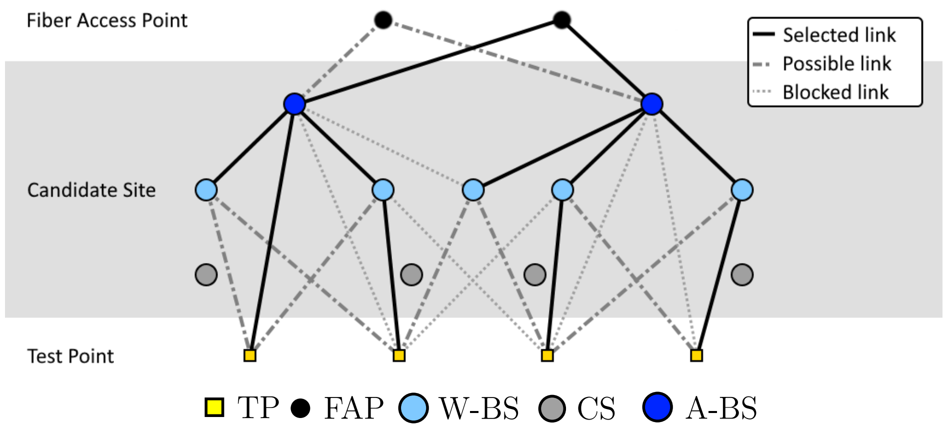

Figure 3.1: System Component Hierarchy

In the remainder of this chapter, we will first describe how we modeled the locations of all system components and their mutual connections. Then, we present our channel model.

3.1 Component locations and possible links

In this section, we describe how we model the system components and their relevant attributes for network deployment. These are also summarized in Table 3.1. The variables that we use to denote the actual deployment are described in Section 4.1.

Let n_{TP} be the number of test points, and let $TP = \{0, 1, .., n_{TP} - 1\}$ be the set of all test points. TPs represent vehicles with roof-mounted antennas. Each $i \in TP$ has a demand of D_i Mbps, and is appointed full demand links to K BSs, of which at least one must be anchored. All BSs are located at candidate sites. Let n_{CS} be the number of candidate sites, and let $CS = \{0, 1, \ldots, n_{CS} - 1\}$ be the set of all candidate sites. Each BS is indicated with its index in CS. The total bandwidth capacity of BS j is denoted as C_j^C.

For each TP, exactly one A-BS is appointed as its primary BS. It provides a link with the lowest latency. All other BSs serving the same TP (including A-BSs) will be wirelessly backhauled by it. This way, the requirement of each TP having a primary BS keeps the latency low, even if a wireless backhaul link is used. In that case, each path is maximized to 2-hop communication, namely from the TP to the W-BS, and then from the W-BS to the primary A-BS.

Access links

To choose CSs for BS deployment, three attributes of each candidate site j are considered. Firstly, the link length between any TP i and a CS j is stored in an $n_{TP} \times n_{CS}$ matrix L. L_{ij} is the euclidean distance between i and j. Secondly, let A be an $n_{TP} \times n_{CS}$ binary coverage matrix that denotes whether LoS communication is possible for an access link between a TP i and a CS j, in which case $A_{ij} = 1$. LoS communication is considered feasible when two conditions are met, namely (1) there is no static blockage (e.g. building) between i and j, and the link length must be below a maximum value, i.e. $L_{ij} < L^{max}$. Lastly, the angle between any two possible links is stored in the $n_{TP} \times n_{CS} \times n_{CS}$ matrix Θ, such that $\Theta_{ijj'}$ is the angle between two LoS links from CSs j and j' to TP i with $j \neq j'$.

Backhaul links

Let n_{FAP} be the number of fiber access points, and let $FAP = \{0, 1, \ldots, n_{FAP} - 1\}$ be the set of all FAPs. The throughput of an FAP is not considered, since this is very likely not the bottleneck. The length of a fiber cable to connect CS j to FAP p is stored in an $n_{CS} \times c_{FAP}$ matrix F. For wireless backhaul links, their length is stored in the $n_{CS} \times n_{CS}$ matrix S. Furthermore, the $n_{CS} \times n_{CS}$ binary coverage matrix B stores whether LoS communication is feasible between CSs, based on static blockages and $S < L^{max}$, similar to A.

Table 3.1: Parameters representing deployment environment data

Parameter	Unit	Size	Description
TP		n_{TP}	Set containing all test points.
CS		n_{CS}	Sets containing all candidate sites.
FAP		n_{FAP}	Sets containing all fiber access points.
i, j, p			Indices used for the elements of the system components: $i \in TP$, $j \in CS$, and $p \in FAP$.
C_j^C	MHz	n_{CS}	The total capacity of each BS j.
D_i	Mbps	n_{TP}	The demand of each TP i.
K		1	The number of full demand links that should be available for each TP.
A_{ij}		$n_{TP} \times n_{CS}$	Indicates whether a LoS link is possible between each combination of TP i and BS j.
L_{ij}	meter	$n_{TP} \times n_{CS}$	The distance between each combination of TP i and CS j.
L^{max}	meter	1	The maximum link length for both access and backhaul links.
$\Theta_{ijj'}$	degree	$n_{TP} \times n_{CS} \times n_{CS}$	The angle between each pair of links from two different CSs j and j' to one TP i.
$B_{jj'}$		$n_{CS} \times n_{CS}$	Indicates whether a LoS link is possible between each pair of BSs j and j'.
$S_{jj'}$	meter	$n_{CS} \times n_{CS}$	The distance between each pair of BSs j and j'.
F_{jp}	meter	$n_{CS} \times n_{FAP}$	The distance between each combination of BS j and FAP p.

3.2 Channel model

Since mmWave channels are noise-limited and are barely influenced by interference ($SNR \approx SINR$) due to highly directional communication [7], we only consider the signal to noise ratio (SNR) value. To calculate the SNR for any link between a CS j and a TP i, the following formula is used:

$$SNR_{ij} = \frac{P_j^{BS} G_j^{TX} G_i^{RX} PL^{-1}(d)}{P_N} \tag{3.1}$$

where P_j^{BS} and G_j^{TX} are the transmit power and antenna transmit gain of BS_j, G_i^{RX} is the antenna receive gain of a user at TP i, $PL(d)$ is the Path Loss over distance d, and P_N is the noise power.

The antenna gain values are the maximal antenna gain, assuming that the transmit and receive beams are perfectly aligned. We adopt the LoS path loss model presented in [26], which is based on the Friis free space path loss model, given by the following formula in dB:

$$PL(d) = \beta \cdot \left(20\ log_{10}\left(\frac{4\pi}{\lambda}\right) + 10\ \alpha\ log_{10}(d)\right) \tag{3.2}$$

where d is the link length, λ is the wave length, α is the path loss exponent, and β is the slope correction factor resulting from fitting the path loss model to the empirical data in [26].

The required bandwidth for providing a given throughput over a given distance is calculated from the SNR value. For this, the Shannon capacity is calculated, and a overhead of 25% is taken into account for control signals, mostly for beam tracking. The formula is as follows for an access link.

$$C_{ij}^a = \frac{D_i}{log_2(1 + SNR_{ij})} \cdot 1.25 \tag{3.3}$$

where C_{ij}^a is the required bandwidth capacity in MHz for a full demand link, D_i is the full demand of TP i in Mbps, and SNR_{ij} is the SNR for a link between a CS j and a TP i as defined in (3.1).

For backhaul links, the capacity C_{ij}^b is calculated likewise. Note the index is ij, and not jj'. This can be done, since each TP must be assigned one primary link to an A-BS, say j'. Therefore, C_{ij}^b is the bandwidth capacity needed for backhaul link between CS j and CS j', in which A-BS j' is the primary BS for TP i.

Chapter 4

Formal description of the optimization problem

In the previous chapter, we have defined the notation of the deployment environment. This chapter presents mmWave CP with MCo and IAB as an optimization problem. Because CP problems are known to be NP-hard [42], the problem will not be solved, but rather be used to define the problem clearly and gain insight to develop a heuristic algorithm in the next chapter. The optimization problem aims to minimize the deployment costs. Since the costs are influenced by the number of W-BSs and not by their locations, the minimal budget can be achieved by multiple different deployments. To choose the best deployment, we introduce another objective function with a lower priority. The secondary objective function aims to optimize for access reliability.

Our algorithm will select CSs to place BSs for serving the TPs, and determine which BSs should connect to which FAPs. Those system components and their connections are used as input, as described in Chapter 3. We define several decision variables and helper variables in this chapter. These are also used in Chapter 5 to explain how our algorithm works.

This chapter is organized as follows. First, Section 4.1 describes the decision variables. Then, we introduce the two objective functions in Sections 4.2 and 4.3. Finally, the constraints are given and the optimization problem is formulated in Section 4.4.

4.1 Decision and helper variables

Below, we first describe the decision variables, which are all binary. These variables are listed in Table 4.1. Then, we explain how the capacity assignment can be derived from these variables. For this, we introduce a number of helper variables. These variables are listed in Table 4.2.

For mmWave CP with MCo and IAB, two issues must be decided on. Firstly, it must be decided where the BSs should be placed. This is tracked in a binary vector o_j, where $o_j = 1$ means that the CS j is occupied by a mmWave BS. This can also be noted as $j \in CS^o$. The set $CS^o \subseteq CS$ consists of all occupied CSs by a BS, whereas $CS^u \subseteq CS$ contains all unoccupied CSs. Therefore, $CS = CS^o \cup CS^u$ and $CS^o \cap CS^u = \varnothing$. Likewise, w_j indicates whether the BS at CS j is provided with a fiber backhaul. Naturally, $w_j = 1$ implies $o_j = 1$. This can also be noted as $j \in CS^w$, where $CS^w \subseteq CS^o$. The set $CS^{\overline{w}} \subseteq CS^o$ contains all W-BSs ($w_j = 0$).

Links between TPs and BSs are stored in the $n_{TP} \times n_{CS}$ binary matrix a_{ij}, where $a_{ij} = 1$ means that a link is selected. Links can only be selected if it is a possible link according to A_{ij}, so $a_{ij} \leq A_{ij}$ must hold. Each TP i has one full demand link to an A-BS j that is referred to as its primary link, and all other links are called secondary links. Primary links are stored in a binary $n_{TP} \times n_{CS}$ matrix p as $p_{ij} = 1$. All other BSs j' serving TP i are wirelessly backhauled to its primary BS j, even if those are anchored as well. This way, latency can be reduced since packets can be rerouted directly by the primary BS when a link gets blocked. For secondary links, backhaul link assignment is recorded in the $n_{CS} \times n_{CS}$ binary matrix b, where $b_{jj'} = 1$ means that BS j' is wirelessly backhauled by primary BS j.

Secondly, it must also be decided which FAPs should be used for A-BSs. For each FAP, a binary vector u_p indicates whether it is used, where $u_p = 1$ means that at least one BS is connected to this FAP. The matrix f denotes backhaul connections between BSs and FAPs, where $f_{jp} = 1$ means that the A-BS at CS j is connected to FAP p with a fiber cable of F_{jp} meter.

A number of helper variables are defined to make sure that all TPs can be served their full demand simultaneously. Bandwidth assignment is stored in $n_{TP} \times n_{CS}$ matrices c^a and c^b for access and backhaul links. Its content is determined as follows. For primary links, c^a_{ij} will be assigned C^a_{ij}, as calculated in (3.3). Since all links are full demand links in our system, the capacity for secondary links is reserved in c^a_{ij} like for primary links. Hence, a_{ij} and c^a_{ij} are directly related as follows: $a_{ij} = 1 \implies c^a_{ij} = C^a_{ij}$.

For backhaul links, bandwidth capacity is reserved similarly. Each BS j' serving TP i ($p_{ij'} = 0, a_{ij'} = 1$) reserves $C^b_{ij'}$ MHz, which is tracked in $c^b_{ij'}$. A primary BS j, however, only reserves the maximum bandwidth needed at any time. Therefore, if no backhaul link bandwidth exceeds c^a_{ij}, c^b_{ij} will be set to 0. Otherwise, the extra bandwidth needed for the longest backhaul link will be set in c^b_{ij}, calculated as follows: $c^b_{ij'} - c^a_{ij}$, $c^b_{ij'} \geq c^b_{ij''} \forall j'' \in CS$.

Table 4.1: Decision variables, all binary

Variable	Size	Description
a_{ij}	$n_{TP} \times n_{CS}$	Indicates whether a possible access link between TP i and CS j is selected.
$b_{jj'}$	$n_{CS} \times n_{CS}$	Indicates whether a possible backhaul link between CS j and CS j' is selected.
f_{jp}	$n_{CS} \times n_{FAP}$	Indicates whether a fiber link between CS j and FAP p is selected. This makes BS j an A-BS ($w_j = 1$) and also implies $u_p = 1$.
o_j	n_{CS}	Indicates whether a CS j is occupied. If $o_j = 1$, this can also be referred to as BS j.
p_{ij}	$n_{TP} \times n_{CS}$	Indicates whether a selected access link between TP i and CS j is the primary link of a TP i.
u_p	n_{FAP}	Indicates whether an FAP j is used, i.e. at least one BS is connected to it.
w_j	n_{CS}	Indicates whether a BS j is anchored, i.e. it is provided with a fiber backhaul link. If $w_j = 1$, BS j is an A-BS.

Table 4.2: Helper variables for capacity assignment and delivered demand

Variable	Unit	Size	Description
c_j^C	MHz	n_{CS}	The used bandwidth capacity of BS j.
c_{ij}^a	MHz	$n_{TP} \times n_{CS}$	The reserved bandwidth capacity at BS j for the access link of TP i.
c_{ij}^b	MHz	$n_{TP} \times n_{CS}$	The reserved bandwidth capacity at BS j for the backhaul link of TP i.
d_i	Mbps	n_{TP}	The total delivered demand over all selected links.

4.2 Deployment costs

The primary objective is to minimize the total network deployment costs. These are calculated from two parts, namely the BS deployment costs and the fiber deployment costs. The expenditures are denoted as variables of the general form E_i^X, where X denotes the type of expenditure, and i is an index or a combination of indices of involved system components. In particular, E_j^D is the deployment cost of a BS at CS j. The deployment costs of BSs at all selected CSs ($o_j = 1$) are therefore:

$$\sum_{j \in CS} o_j E_j^D \tag{4.1}$$

The costs of connecting BSs with fiber cables to FAPs is calculated from three elements:

1. The costs of installation of an optical splitter for connecting to FAP p is E_p^F, with which the costs of all used FAPs ($u_p = 1$) can be calculated as follows:

$$\sum_{p \in FAP} u_p E_p^F \tag{4.2}$$

2. The costs of the fiber deployment between the CS j and FAP p is determined by two factors. First, the costs of the cable per length unit, denoted as E^C. Then, this must be multiplied by the distance F_{jp}. The costs of all cables to be deployed between any selected FAP p and A-BS j ($f_{jp} = 1$) are therefore:

$$\sum_{j \in CS} \sum_{p \in FAP} f_{jp} E^C F_{jp} \tag{4.3}$$

3. The additional costs of the hardware (e.g. a modem) for connecting an A-BS at CS j ($w_j = 1$) to a fiber compared to a W-BS is E^A. This leads to the expression:

$$\sum_{j \in CS} w_j E^A \tag{4.4}$$

The total expenditure of network deployment is therefore:

$$E_{tot} = \sum_{j \in CS} o_j E_j^D + \sum_{p \in FAP} u_p E_p^F + \sum_{j \in CS} \sum_{p \in FAP} f_{jp} E^C F_{jp} + \sum_{j \in CS} w_j E^A$$

Which can be rewritten as follows.

$$E_{tot} = \sum_{j \in CS} \left(o_j E_j^D + w_j E^A \right) + \sum_{p \in FAP} u_p E_p^F + \sum_{j \in CS} \sum_{p \in FAP} f_{jp} E^C F_{jp} \tag{4.5}$$

In this formula, all expenses of network deployment are summed in this order: deployment costs for BSs and fiber connections, the utilization of FAPs, and the costs for connecting A-BSs to FAPs.

4.3 Blockage sensitivity

The secondary objective function aims to maximize the access reliability under dynamic blockages as in [8]. We will express it as blockage sensitivity, which will be minimized. To describe and quantify this, two network characteristics are used as input, namely the angular diversity (AD) and the average link length (LL) for each TP, as elaborated below. A simple example scenario is presented in Figure 4.1. This approach has been proved effective by Devoti et al. [8]. In their study, they used randomly placed dynamic obstacles of small size. Therefore, the angular diversity was used to counter self-blockage, and link length to reduce the chance of an obstacle intervening the link. Since we consider roof-mounted antennas on vehicles, the LL argument still holds. Angular diversity, however, serves another purpose, namely to decrease the probability for large moving obstacles like trucks and buses to obstruct multiple links simultaneously. How values for these metrics are calculated in [8] is explained below.

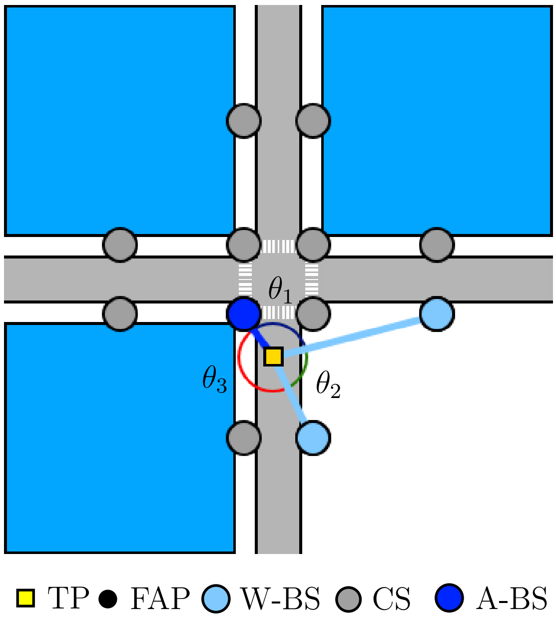

Figure 4.1: TPs with 3-connectivity

Angular Diversity: The AD score is denoted as R_A. Its value is determined using the minimum angle δ_i between any two selected links for a TP. δ_i is defined as follows.

$$\delta_i \leq \Theta_{ijj'} + 2\pi(2 - a_{ij} - a_{ij'}) \quad \forall j, j' \in CS, \; \forall i \in TP$$

For example, in Figure 4.1 δ_i would be θ_2. Note that the angles between non-selected BSs for a TP are de-activated using a big-M technique. This works as follows. Only when both BSs j and j' are serving TP i, $a_{ij} = a_{ij'} = 1$ holds. In any other case, the angle is dismissed as possible candidate for smallest angle by adding one or two times 2π.

R_A, the normalized average of all values for δ_i, is calculated as follows:

$$R_A = \frac{1}{n_{TP}} \sum_{i \in TP} \frac{\delta_i}{M} \tag{4.6}$$

where the constant $M = \frac{2\pi}{K}$ is the maximum possible value for δ_i, to make R_A assume a value in $[0, 1]$. This value should be maximized to reduce the blockage sensitivity.

Average Link Length: The LL score is denoted as R_L. This is determined by the average length of all $n_{TP} \cdot K$ links in the system. The average link length for a TP i, ℓ_i, is calculated as follows.

$$\ell_i = \frac{1}{K} \sum_{j \in CS} a_{ij} L_{ij} \quad \forall i \in TP$$

where a_{ij} selects only the lengths of the used links.

The normalized average of all link lengths R_L can then be calculated as follows:

$$R_L = \frac{1}{n_{TP}} \sum_{i \in TP} \frac{\ell_i}{L^{max}} \tag{4.7}$$

where the value of ℓ_i is normalized by the the maximum link length parameter L^{max} to make R_L assume a value in $[0, 1]$. This value should be minimized to reduce the blockage sensitivity.

Combined: These two functions can be combined in one objective function to be minimized. To accomplish this, the value for angular diversity (which should be maximized) must be inverted by subtracting it from 1. Furthermore, the auxiliary variable χ is introduced to set the balance between the two blockage sensitivity metrics. This results in:

$$R_{AL} = \frac{1}{n_{TP}} \sum_{i \in TP} \left(\chi \cdot \frac{\ell_i}{L^{max}} + (1 - \chi) \cdot \left(1 - \frac{\delta_i}{M} \right) \right) \tag{4.8}$$

where χ must be set to a value in $[0, 1]$.

4.4 Optimization problem formulation

Our algorithm will deploy a mmWave network while trying to minimize the costs (E_{tot}) and then to also maximize the access reliability (R_{AL}). Both objectives have been formalized in Section 4.2 and Section 4.3, respectively. In this section, further constraints for our system are explained and formalized in (4.9b) - (4.9o).

The constraints can be divided in four categories according to their purpose: (1) Defining backhaul links between BSs and FAPs, eventually relayed over A-BSs. (2) Defining user links from BSs to TPs, and (3) Constraining capacity assignment to what is available. (4) Define the domains of all decision variables.

(See next page for the descriptions of all constraints side-by-side to the optimization problem.)

Backhaul links

- Constraint (4.9b) makes sure that a fiber backhaul only connects to a CS ($w_j = 1$) at which a BS is placed ($o_j = 1$).
- Constraint (4.9c) limits wireless backhaul links to links between a deployed A-BS ($w_j = 1$) and a deployed BS ($o'_j = 1$), between which a LoS link is possible ($B_{jj'} = 1$).
- Constraints (4.9d) and (4.9e) force each fiber backhaul f_{jp} to connect an A-BS to a used FAP ($u_p = 1$), and let each A-BS be connected to exactly one FAP.
- Constraint (4.9f) demands that the secondary BSs j' of each TP i can be LoS-backhauled by its primary BS j.

Access links

- Constraint (4.9g) requires the number of links for each TP i to be equal to K.
- Constraint (4.9h) limits the selection of active links ($a_{ij} = 1$) for each TP to those that provide a LoS connection to a deployed BS ($o_j = 1$).
- Constraints (4.9i) and (4.9j) define the primary link of each TP to be a link to an A-BS ($w_j = 1$) and limit the number of primary links to 1.

Capacity assignment

- Constraint (4.9k) requires that each BS j serving a TP i reserves bandwidth capacity for a full demand access links.
- Constraint (4.9l) demands that sufficient backhaul bandwidth is reserved for secondary BSs. A secondary BS j serving TP i means that $a_{ij} = 1 \wedge p_{ij} = 0$. In that case, the backhaul bandwidth for a full demand link (C^b_{ij}) should be reserved. In all other cases, namely for primary BSs and unoccupied CSs, this constraint sets the minimum reserved bandwidth to 0.
- Constraint (4.9m) demands that sufficient bandwidth is reserved at primary base stations for the longest backhaul link, since only one path will be active at any time. The primary BS j of TP i has $p_{ij} = 1$. In that case, the extra bandwidth with respect to c^a_{ij} needed for the longest backhaul link to any secondary BS j' should be reserved in c^b_{ij}. When all backhaul links need less bandwidth than the primary link (c^a_{ij}), the right hand value of this constraint will be negative. However, constraint (4.9l) then limits the extra reserved bandwidth to be at least 0, as noted above. The same holds for this constraint: when $p_{ij} = 0$, the right hand side will be 0.
- Constraint (4.9n) requires the reserved capacity for both access and backhaul links for each BS j to never exceed its available bandwidth C^C_j.

Domains of decision variables

- Constraint (4.9o) defines all decision variables to be binary.

Finally, it should be noted that the objective functions are given in order of priority. This means that the system costs E_{tot} should be minimized first, and then the AD&LL-score R_{AL} should be minimized without introducing extra costs. The optimization problem is given as follows.

$$
\begin{aligned}
\underset{a_{ij},\, p_{ij},\, b_{jj'},\, f_{jp},\, o_j,\, w_j,\, u_p}{\text{minimize}} \quad & [E_{tot}, R_{AL}] && && \text{(4.9a)}\\
\text{subject to} \quad & && && \\
w_j \leq o_j \quad & && \forall j \in CS, && \text{(4.9b)}\\
b_{jj'} \leq w_j o_{j'} B_{jj'} \quad & && \forall j, j' \in CS : j \neq j', && \text{(4.9c)}\\
f_{jp} \leq u_p \quad & && \forall j \in CS, \forall p \in FAP, && \text{(4.9d)}\\
\sum_{p \in FAP} f_{jp} = w_j \quad & && \forall j \in CS, && \text{(4.9e)}\\
a_{ij'} \leq p_{ij} b_{jj'} \quad & && \forall i \in TP, && \text{(4.9f)}\\
& && \forall j, j' \in CS : j \neq j', && \\
\sum_{j \in CS} a_{ij} = K, \quad & && \forall i \in TP, && \text{(4.9g)}\\
a_{ij} \leq o_j A_{ij} \quad & && \forall i \in TP, \forall j \in CS, && \text{(4.9h)}\\
p_{ij} \leq w_j a_{ij} \quad & && \forall i \in TP, \forall j \in CS, && \text{(4.9i)}\\
\sum_{j \in CS} p_{ij} = 1 \quad & && \forall i \in TP, && \text{(4.9j)}\\
c^a_{ij} = a_{ij} C^a_{ij} \quad & && \forall i \in TP, \forall j \in CS, && \text{(4.9k)}\\
c^b_{ij} \geq a_{ij}(1 - p_{ij}) C^b_{ij} \quad & && \forall i \in TP, \forall j \in CS, && \text{(4.9l)}\\
c^b_{ij} \geq p_{ij} \cdot (c^a_{ij} - b_{jj'} C^b_{ij'}) \quad & && \forall i \in TP, \forall j, j' \in CS, && \text{(4.9m)}\\
C^C_j \geq \sum_{i \in TP} c^a_{ij} + c^b_{ij} \quad & && \forall j \in CS, && \text{(4.9n)}\\
a_{ij}, b_{jj'}, p_{ij}, f_{jp}, u_p, o_j, w_j \in \{0, 1\}, \quad & && \forall i \in TP, \forall p \in FAP, && \text{(4.9o)}\\
& && \forall j, j' \in CS : j \neq j' &&
\end{aligned}
$$

Chapter 5

Introducing the MIND-GO algorithm

This chapter presents our proposed approximation algorithm for mmWave BS deployment, considering their backhaul types. We name it the MCo with IAB mmWave Network Deployment using Greedy construction with Optimization (MIND-GO) algorithm. The former chapter presented the optimization problem to make the two objectives and all the constraints clear. With the MIND-GO algorithm, we intend to approach the optimal solution with polynomial computation complexity, which is the best achievable. This is known to be possible with greedy construction (GC) algorithms [10].

For GC, all CSs are scored, and a BS is deployed at the highest scoring CS. This is repeated until all TPs are sufficiently covered. The two phases that place new BSs apply this approach. Additionally, three optimization phases are included to obtain a better solution, while keeping the complexity polynomial. Below, we will first describe the five phases of the MIND-GO algorithm on a high level. Then, we will briefly recapitulate the parameters and variables as presented in the previous chapters. Finally, we will describe each phase in more detail and determine its computational complexity.

The MIND-GO algorithm consists of the following five phases.

1. Use GC to select a subset of CSs for A-BS placement, providing sufficient capacity for all TPs for full demand access and wireless backhaul links.

2. Remove BSs that are not needed and select the shortest links as primary.

3. Use GC to select a subset of unoccupied CSs for Wireless BS placement to provide multi-connectivity to all TPs.

4. Swap used links with unused links as long as this reduces the blockage sensitivity.

5. Try to reduce the system costs by removing as much BSs as possible.

Throughout all these phases, the aim is to minimize the objective functions for costs (E_{tot}) and AD&LL-score (R_{AL}). This is mainly achieved by the final two phases. The aims of the five phases are as follows. Phase 1 deploys more BSs than needed, such that the best locations of A-BSs for primary links can be selected in Phase 2. Phase 2 namely removes A-BSs that only cover TPs that can already be served by other A-BSs and then selects one primary link for each TP. Phase 3 then deploys W-BSs for K-connectivity and assigns only full demand links to TPs, selecting links with the best AD&LL score first. Phase 4 improves the AD&LL score to make sure the best BSs are not removed in the final phase. This phase does not influence the system costs, but only alters the link assignment. Finally, Phase 5 first removes as much fiber backhaul cables as possible, and then as much W-BS as possible. This is tried for each BS, starting with the one with the lowest number of links. This way, the costs are reduced at the cost of AD&LL score, but due to Phase 4 this score is kept as high as possible. In Algorithm 1, an overview is given of which sub-algorithms are executed in each phase.

Algorithm 1: BS Deployment: main algorithm

Input : TP, CS, FAP, A, L, Θ, B, C^C, D, F, C^a, C^b, C^{max}, χ, Γ^A, Γ^W

Output: a, p, b, f, and their derivatives o, w and u

Phase 1: Initial deployment of A-BSs for 1-connectivity

while *any TP i has no full demand link:* $\exists i \in TPP, \forall j \in CS : a_{ij} < 1$ **do**

  run Algorithm 1.1a and 1.1b

Phase 2: Remove A-BSs and select primary links

run Algorithm 1.2a and 1.2b

Phase 3: Initial deployment of W-BSs for K-connectivity

run Algorithm 1.3

Phase 4: Swap links to optimize AD&LL-score

run Algorithm 1.4

Phase 5: Remove obsolete BSs to reduce costs

run Algorithm 1.5a and 1.5b

The inputs for the MIND-GO algorithm can be divided into four categories. First, the three sets of the system components (TP, CS and FAP) are needed.

Next, there are the matrices A, L, Θ, B, C^C, D, and F, defining the scenario, as described in Chapter 3. These are summarized in Table 3.1. Note that S is not used as input for the algorithm. Its value is namely only used to calculate the needed backhaul link capacity C^b, as described below. The access link length L, however, is used as input since it is needed to calculate the AD&LL score, combined with Θ.

Additionally, the required bandwidths for full demand links are supplied to the algorithm with C^x-names. These are calculated from the wireless link lengths (in L and S), as explained in Section 3.2 (see formula (3.3) in particular). The notation and use of these variables is summarized in Table 5.1. The d and c^x variables are not listed as in- and output variables for the sub-algorithms, since they can be calculated from a and the D/C^x parameters.

Table 5.1: Link capacity values in MHz as calculated from map parameters

Constant	**Variable**	**Description**
C_{ij}^a	c_{ij}^a	C^a represents the needed bandwidth capacity for a full demand access link (D_i) from BS j to TP i, c^a represents the used capacity, and equals $a_{ij} \cdot C_{ij}^a$ or C_i^{max}.
C_{ij}^b	c_{ij}^b	C^b represents the needed bandwidth capacity for a full demand backhaul link from BS j to the primary base station (P-BS) of TP i, c^b represents the used capacity (see Section 4.4, explanation of (4.9l) and (4.9m)).
C_i^{max}	c_{ij}^a	C^{max} is the bandwidth capacity needed for a full demand link (D_i) of TP i when the link length equals L^{max}, c^a is set to this value in the first two phases to make sure enough bandwidth can be assigned to backhaul links in Phase 3.

The final three parameters are vectors containing scoring weights. The first of these is χ to set the balance between AD and LL in the AD&LL score. The other two, Γ^A and Γ^W, define the weights of the partial scores of CSs for respectively A-BS and W-BS deployment. These are detailed in the descriptions of phases 1 and 3 below.

The decision variables of the MIND-GO algorithm are explained in Chapter 4. These are summarized there in Table 4.1. These variables are used as input for some sub-algorithms, because they build on the results of former algorithm steps.

Now, we will discuss the 5 phases in more detail.

5.1 Initial deployment of A-BSs for 1-connectivity

In the first phase, A-BSs are deployed for 1-coverage. This is done in two steps. The first step is to deploy sufficient A-BSs that jointly provide the needed capacity for all TPs. The pseudo-code for this is given in *Algorithm 1.1a*. This step results in at least one full demand link for each TP to an A-BS, denoted by a, and the fiber links from those A-BSs to FAPs, denoted by f. From those variables, the values of o, w and u can be derived. The input needed for this step includes the sets of system components. Furthermore, the matrices A and F are needed to decide where to place the A-BSs, as well as the lists with the available capacity at each BS (C^C) and the needed capacity for full demand links (C^{max}).

As said before, a greedy approach is applied in this step. For this, each CS is scored at line 2. These scores (z) are based on three properties as listed below.

1. The length of the fiber cable to the closest FAP, divided by the longest fiber cable length. This value is inverted by subtracting it from 1.

2. The number of TPs within reach according to A that are not served their full demand ($d_i < D_i$), divided by the resulting maximum.

3. The average AD&LL score for all those TPs.

Each of these result in a value in $[0, 1]$ and is assigned a weight in the parameter Γ^A. The sum of weights is 1, so the weighed sum of partial scores is also on the scale $[0, 1]$.

Next, at lines 4-6 an A-BS is placed at the highest scoring CS. Then, its available bandwidth is divided equally over the covered TPs at lines 7-16. If this results in any TP being served more bandwidth than its demand (t^{js}), the surplus bandwidth c^s is divided among all TPs that are not assigned a full demand link to the newly placed CS j (t^{jr}). Note that some bandwidth may remain unused if $t^{jr} = \varnothing$ but $t^{js} \neq \varnothing$. Finally, the link size relative to full demand is calculated and the newly delivered demand is added to d_i at lines 17-18. Note that the matrix a stores non-binary values in this step. From Phase 2 and onward, a will only store binary values conform the system model.

Algorithm 1.1a: Deploy A-BSs until all TPs are sufficiently covered

Input : TP, CS, FAP, A, L, Θ, F, C^C, C^{max}, χ, Γ^A

Output: a, f, and their derivatives o, w and u

while *any TP i is not served its full bandwidth:* $\exists i \in TP : d_i < D_i$ **do**

  let z be the scores for A-BS placement for each CS $j \in CS^u$

  let $j \in CS^u$ be the CS with the highest score in z: $z_j \leq z_{j'} \forall j' \in CS^u$

  deploy an A-BS at CS $j \in CS^u$: $o_j \leftarrow 1, w_j \leftarrow 1$

  remove j from CS^u, add it to CS^o: $CS^u \leftarrow CS^u \setminus \{j\}$, $CS^o \leftarrow CS^o \cup \{j\}$

  connect j with the nearest FAP p: $f_{jp} \leftarrow 1,\ S_{jp} \leq S_{jp'}\ \forall p' \in FAP$

  let $t \subseteq TP$ be all TPs that have some remaining demand: $t \leftarrow \{i \in TP |\, d_i < D_i\}$

  let $t^j \subseteq t$ be all TPs within LoS of CS j: $t^j \leftarrow \{i \in t |\, A_{ij} = 1\}$

  divide the bandwidth of CS j equally among TPs in t^j: $c^a_{ij} \leftarrow \frac{C^C_j}{|t^j|}\ \forall i \in t^j$

  let $t^{js} \subseteq t^j$ be all TPs with more bandwidth from CS j than needed for their demand at maximum link length: $t^{js} \leftarrow \{i \in t^j |\, c^a_{ij} > C^{max}_i\}$

  while $t^{js} \neq \varnothing$ **do**

    calculate the surplus bandwidth: $c^s \leftarrow \sum_{i \in t^{js}} \left(c^a_{ij} - C^{max}_i\right)$

    let $t^{jr} \subset t^j$ be all TPs with less bandwidth from CS j than needed for their demand: $t^{jr} \leftarrow \{i \in t^j |\, c^a_{ij} < C_{ij}\}$

    remove the surplus bandwidth from the TPs: $c^a_{ij} \leftarrow C^{max}_i\ \forall i \in t^{js}$

    divide the surplus bandwidth c^s equally over all TPs in t^{jr}: $c^a_{ij} \leftarrow c^a_{ij} + \frac{c^s}{|t^{jr}|}\ \forall i \in t^{jr}$

    determine which TPs have surplus bandwidth now: $t^{js} \leftarrow \{i \in t^{jr} |\, c^a_{ij} > C^{max}_i\}$

  calculate the link size: $a_{ij} \leftarrow \frac{c^a_{ij}}{C^{max}_i}\ \forall i \in t^j$

  update total delivered demand: $d_i \leftarrow d_i + a_{ij} \cdot D_i\ \forall i \in t^j$

The complexity of the first step is $O(n_{CS} \times n_{FAP} \times n_{TP} + n_{CS} \times n_{TP}^2)$. This is mainly the result of the score calculation for each CS. The three partial scores for each CS are either determined by its links to FAPs or by its links to TPs This results in a complexity of $O(n_{CS} \times n_{FAP} + n_{CS} \times n_{TP})$. Since the score is recalculated after an A-BS is placed, this has to be multiplied by n_{TP}. For each TP namely at most one A-BS must be deployed, in the worst case. Therefore, the number of placed A-BSs is constrained by n_{TP}.

The second step of Phase 1 is to form as much full demand links as possible. This is detailed in *Algorithm 1.1b*. This step results in a binary a matrix, since only full demand links are created. For this, entirely new links will be created between the placed A-BSs and the TPs. Therefore, only o is needed as input variable, in addition to some system parameters.

This step has a complexity of $O(n_{TP}^2)$, because for each placed A-BS, its covered TPs are provided a full demand link. The number of A-BSs is constrained by n_{TP}, as noted above.

The reorganization of links is done for each deployed A-BS. For this, first a set o^j of all A-BSs covering the same TPs in t^j is composed (lines 2-5). Then, a full demand link is assigned to each covered TP, every time choosing the shortest links of all covering A-BSs. The newly covered TP is subsequently removed from the set t^j at line 11. When an A-BS has not enough bandwidth available, it is removed from the set o^j at line 13.

Some TPs might be unconnected after this step has been executed because the first step of this phase only guarantees enough total bandwidth. Hence, only links for partial demand might be available for some TPs. If this is the case, new A-BSs need to be placed using Algorithm 1.1a. Therefore, both steps of this phase are repeated until all TPs have at least one full demand link, as can be seen at lines 1-2 of Algorithm 1. This loop is executed two times at most, since only a few TPs will be left uncovered after the first run. Hence, this loop does not contribute to the big-O complexity. The complexity of this phase is $O(n_{CS} \times n_{FAP} \times n_{TP} + n_{CS} \times n_{TP}^2)$, since this is the maximum complexity of both steps.

Algorithm 1.1b: Reorganize links to form as much full demand links to A-BSs as possible

Input : TP, CS, A, C^C, C^{max}, o
Output: a

for *each deployed BS:* $j \in CS^o$ **do**
  let $t^j \subseteq TP$ be all TPs connected to BS j: $t^j \leftarrow \{i \in TP |\, a_{ij} > 0\}$
  only keep all TPs in t^j that have no full demand link yet:
  $t^j \leftarrow \{i \in t^j |\, \nexists j' \in CS^o : a_{ij'} = 1\}$
  let $o^j \subseteq CS^o$ be all BSs serving the TPs in t^j (including j):
  $o^j \leftarrow \{j' \in CS^o |\, a_{ij'} > 0\, \forall i \in t^j\}$
  remove links to all covered TPs from all their serving BSs:
  $a_{ij'} \leftarrow 0, c^a_{ij'} \leftarrow 0\, \forall i \in t^j, j' \in o^j$
  while *any covered TP has no full demand link and bandwidth is available:*
  $t^j \neq \varnothing \wedge o^j \neq \varnothing$ **do**
    for *each covering BS:* $j' \in o^j$ **do**
      choose the closest TP i: $i \in t^j : \forall i' \in t^j : L_{ij'} \leq L_{i'j'}$
      if *BS* j' *has enough bandwidth to serve TP* i*:* $c^C_{j'} + C^{max}_i \leq C^C_{j'}$ **then**
        make a full demand link for TP i: $a_{ij'} \leftarrow 1$, $c^a_{ij'} \leftarrow C^{max}_i$
        remove selected TP i from TPs yet to serve: $t^j \leftarrow t^j \setminus \{i\}$
      else
        remove BS j' from possible serving BSs: $o^j \leftarrow o^j \setminus \{j'\}$
  calculate new values for c^C for all BSs in o^j: $\forall j' \in o^j : c^C_{j'} \leftarrow \sum_{i \in t^j} \left(c^a_{ij'}\right)$

5.2 Remove A-BSs and select primary links

The second phase removes obsolete A-BSs and selects one primary link for each TP. These two steps are explained below. *Algorithm 1.2a* presents the first step. It removes A-BSs that are obsolete. An A-BS is called obsolete if all links can be removed without breaking any constraint. This means that all covered TPs have at least one other full demand link. In this step, a and f (and their derivatives) are altered as described below. The resulting values of a and f from Phase 1 are taken as inputs.

To identify obsolete A-BSs, first all unused capacity of placed A-BSs is used to form full demand links to TPs within reach according to A that are not yet covered according to a. Short links are preferred over longer links. The pseudo-code for this is provided at lines 1-7. Hereby, the chance for obsolete A-BSs is enlarged. Next, the obsolete A-BSs are removed at lines 8-15. As long as there are obsolete A-BSs, the one with the lowest score is removed. The complexity of this step is equal to that of the first step in Phase 1, because the A-BSs are scored after one is removed. Hence the complexity of this step is $O(n_{CS} \times n_{FAP} \times n_{TP} + n_{CS} \times n_{TP}^2)$.

In the second step, a primary link is selected for each TP. This is presented in *Algorithm 1.2b*. The results are stored in p, and other links are removed from a. Little input data is needed: only the link lengths in L and the already assigned links in a. In this step, first the shortest link of each TP is selected as primary link at lines 1-3. This has a complexity of $O(n_{TP}^2)$, because for each TP its links to all connected A-BSs are compared to find the shortest link. Then, all other links are removed at lines 4-6. This has a complexity of $O(n_{TP}^2)$ as well, since all links are checked. The overall complexity of this phase is determined by step 1, which is $O(n_{CS} \times n_{FAP} \times n_{TP} + n_{CS} \times n_{TP}^2)$.

Algorithm 1.2a: A-BS Deployment: Remove Obsolete A-BSs

Input : TP, CS, A, L, Θ, C^C, D, F, C^{max}, χ, Γ^A, a, f, o, u, w

Output: a, f, and their derivatives o, w and u

1 **for** *all placed A-BSs:* $j \in CS^o$ **do**

2   let t^{ju} be all uncovered TPs within LoS: $t^{ju} \leftarrow \{i \in TP |\ A_{ij} = 1 \wedge a_{ij} = 0\}$

3   **while** *unused capacity and uncovered TPs are available:* $c_j^C < C_j^C \wedge t^{ju} \neq \varnothing$ **do**

4   let i be the closest uncovered TP within LoS: $L_{ij} \leq L_{i'j} \forall i' \in t^{ju}$

5   assign as much bandwidth as possible to TP i, but never more than needed for its full demand: $c_{ij}^a \leftarrow min\left(C_i^{max}, C_j^C - c_j^C\right)$

6   add used capacity to c_j^C: $c_j^C \leftarrow c_j^C + c_{ij}^a$

7   remove newly covered TP from t^{ju}: $t^{ju} \leftarrow t^{ju} \setminus \{i\}$

8 let o^o be the set of obsolete A-BSs: $o^o \leftarrow \{j \in CS^o |\ \forall i \in TP\ \exists j' \in CS^o : a_{ij} = 1 \Rightarrow a_{ij'} = 1\}$

9 **while** *there is an obsolete A-BS:* $o^o \neq \varnothing$ **do**

10   let z be the deployment score for all deployed A-BSs in CS^o

11   let j be the obsolete A-BS with the lowest score: $z_j \leq z_{j'} \forall j' \in o^o$

12   remove all links from A-BS j: $a_{ij} \leftarrow 0, c_{ij}^a \leftarrow 0\ \forall i \in TP$

13   remove A-BS j itself: $w_j \leftarrow 0, o_j \leftarrow 0$

14   remove j from CS^o, add it to CS^u: $CS^o \leftarrow CS^o \setminus \{j\}$, $CS^u \leftarrow CS^u \cup \{j\}$

15   update o^o like above (line 8)

Algorithm 1.2b: A-BS deployment: Select Primary Links

Input : TP, CS, L, a

Output: a, p, and their derivative o

1 **for** *each TP:* $i \in TP$ **do**

2   choose the closest connected A-BS j: $a_{ij} = 1 \wedge L_{ij} \leq L_{ij'}\ \forall j' \in CS^o : a_{ij'} = 1$

3   select the link to A-BS j as primary: $p_{ij} \leftarrow 1$

4 **for** *all links:* $i \in TP, j \in CS^o$ **do**

5   **if** $a_{ij} = 1 \wedge p_{ij} = 0$ **then**

6   remove the link: $a_{ij} \leftarrow 0, c_{ij}^a \leftarrow 0$

5.3 Initial deployment of W-BSs for K-connectivity

The third phase deploys W-BSs for multi-connectivity. This is presented in *Algorithm 1.3*. In this phase, each TP is provided with K full demand links in total, including the already assigned primary link. The new links are represented in a, and for newly placed W-BSs a 1 is stored o. To choose W-BSs locations, the current links in a ($a = p$) and possible access and backhaul links in A and B are needed as input. Furthermore, the needed bandwidth capacities for all possible links are provided in C^a and C^b. Additionally, the parameters L and Θ are needed to calculate the AD&LL score, which is weighed by χ. Finally, the parameter Γ^W provides the weights of the two partial scores for each CS as listed below.

1. The number of TPs within reach according to A that are not covered K times ($k_i < K$), divided by the resulting maximum.
2. The average AD&LL score for all those TPs.

Each of these result in a value in $[0, 1]$. The sum of weights in Γ^W is 1, so the weighed sum of partial scores is also on the scale $[0, 1]$.

This score is calculated at line 2, and the CS with the highest score is selected on the next line. It should be noted that not only unoccupied CSs are scored, since it is very well possible that a deployed A-BS has some remaining capacity. This can then be used to provide a TP with a secondary link. If the selected CS is unoccupied, a new W-BS is deployed at lines 4-6. Next, the TPs that are within reach of the selected BS are determined at lines 7-9. Those are sorted on AD&LL score in descending order. These are the scores for each TP if the selected BS would serve it. Then, links are added to the selected BS as long as its capacity permits it at lines 10-19. We recall that A-BSs have reserved bandwidth capacity for the TPs they serve for the maximum link length (L^{max}). When all served TPs are covered by K BSs, the bandwidth capacity that is allocated for access and backhaul links can be updated to the amount that is actually needed. This is done at lines 21-25. This way, the allocated bandwidth (c^C) is reduced, enabling it to be used for secondary links to other TPs. These links might be changed to primary links in phase 5.

The highest complexity in this phase is achieved by the calculation of the AD&LL scores at line 2 and line 9. The complexity thereof is $O(K \times n_{CS} \times n_{TP})$, since the two partial scores for each CS are determined by its possible links to TPs within reach. This is then repeated until all TPs are covered by K BSs. These lines are executed for each TP, resulting in a complexity of $O(K \times n_{CS} \times n_{TP}^2)$.

Algorithm 1.3: W-BS Deployment for K-coverage

Input : TP, CS, a, p, o, A, B, C^C, C^a, C^b, L, Θ, χ, Γ^W, K

Output: a and b, and their derivative o

while *any TP i is not K-covered: $\exists i \in TP : k_i < K$* **do**

  let z be the scores for W-BS placement for each CS $j \in CS$

  let $j \in CS$ be the CS with the highest score in z: $z_j \leq z_{j'} \forall j' \in CS$

  if *no BS deployed at CS j: $o_j = 0$* **then**

    deploy a W-BS at CS $j \in CS^u$: $o_j \leftarrow 1, w_j \leftarrow 1$

    remove j from CS^u, add it to CS^o: $CS^u \leftarrow CS^u \setminus \{j\}$, $CS^o \leftarrow CS^o \cup \{j\}$

  let $t \subseteq TP$ be all TPs that have are not fully covered: $t \leftarrow \{i \in TP | \, k_i < K\}$

  let $t^{ju} \subseteq t$ be all uncovered TPs within LoS of CS j: $t^{ju} \leftarrow \{i \in t | \, A_{ij} = 1 \wedge a_{ij} = 0\}$

  sort t^{ju} based on the AD&LL link score of each link from CS j to any $i \in t^{ju}$

  while *unused capacity and uncovered TPs are available: $c^C_j < C^C_j \wedge t^{ju} \neq \varnothing$* **do**

    let i be the first TP in t^{ju}, with the highest score

    if *available capacity is insufficient for a full demand link: $C^C_j - c^C_j < C^a_{ij}$* **then break**

    assign bandwidth for a full demand user link to TP i: $c^a_{ij} \leftarrow C^a_{ij}$

    assign bandwidth for the backhaul link to TP i: $c^b_{ij} \leftarrow C^b_{ij}$

    add used capacity to c^C_j: $c^C_j \leftarrow c^C_j + c^a_{ij} + c^b_{ij}$

    remove newly covered TP from t^{ju}: $t^{ju} \leftarrow t^{ju} \setminus \{i\}$

    set the link size: $a_{ij} \leftarrow 1$

    update total delivered demand: $d_i \leftarrow d_i + D_i$

    update the coverage: $k_i \leftarrow k_i + 1$

  let $t^c \subseteq t$ be all TPs that are just now fully covered: $t^c \leftarrow \{i \in t | \, k_i = K\}$

  for *all newly fully covered TPs: $i \in t^c$* **do**

    let j be the P-BS of TP i: $p_{ij} = 1$

    remove the former reserved bandwidth: $c^C_j \leftarrow c^C_j - C^{max}_i$

    update the user link bandwidth: $c^a_{ij} \leftarrow C^a_{ij}$

    set the total reserved bandwidth to the maximum of all backhaul links and the user link of TP i: $c^C_j \leftarrow c^C_j + max\left(\{c^a_{ij}\} \cup \{c^b_{ij'} \forall j' \in CS^o\}\right)$

5.4 Swap links to optimize AD&LL-score

In the fourth phase, the AD&LL score is enhanced until no improvement is possible. The pseudo-code for this is presented in *Algorithm 1.4*. This results in changed access and backhaul links, reflected in the values of a and b. The inputs needed for this are the assigned links from the previous phase as represented in a, p, and b. It must also be known which other links are possible, hence A and B are passed as parameter. Furthermore, the demands are given as input with C^a and C^b, as well as the total available capacity of each BS in C^C. Finally, the values of L, Θ and χ are needed to calculate the AD&LL scores.

First, the AD&LL score improvement is calculated for each link and all alternative links at lines 1. Then, links of TPs are swapped to other BSs as long as this results in a higher score at lines 2-12. Each iteration consists of three parts. First, the involved TP and BSs are determined at lines 3-5. Then, the links resulting in the highest score gain are swapped at lines 6-11. At the end of each iteration (line 12), the AD&LL scores are updated, but only for alternative links to TP i. This results in the same scores as when they would be calculated anew, but with linear complexity instead of squared.

In this phase, the highest complexity is achieved again in the lines for score calculation. We recall that the number of placed A-BSs is assumed to be of order $O(n_{TP})$. The number of placed W-BSs is similarly assumed to be of order $O(K \times n_{TP})$. Because all alternative links (to all BSs) of each TP are considered, this results in a complexity of $O(K^2 \times n_{TP}^2)$. Recalculating the scores is done with linear complexity $O(K \times n_{TP})$, but this is repeated until all TPs have optimal links, resulting in the same complexity as the initial score calculation. This is assumed to be in order $O(K \times n_{TP})$, since all links could then be swapped. Therefore, the overall complexity of this phase is determined as $O(K^2 \times n_{TP}^2)$.

Algorithm 1.4: AD&LL Optimization for secondary links

Input : TP, CS, a, p, b, A, B, C^C, C^a, C^b, L, Θ, χ

Output: a, b

1 let $z_{ijj'}$ be the AD&LL score improvements for all the used secondary links from a TP i to a CS j, when it is replaced by the unused but possible link to CS j' (with sufficient bandwidth available)

2 **while** *improvement is possible:* $min\,(z_{ijj'}) > 0$ **do**

3 let i, j be the TP and BS connected by the link with the highest score improvement in $z_{ijj'}$

4 let j' be the BS to which TP i can be connected for this improvement

5 let j^p be the P-BS of TP i: $p_{ij} = 1$

6 remove reserved bandwidth from the P-BS of TP i: $c^C_{j^p} \leftarrow c^C_{j^p} - max\left(\{c^a_{ij^p}\} \cup \{c^b_{ij'} \forall j' \in CS^o\}\right)$

7 remove the reserved bandwidth at BS j: $c^C_j \leftarrow c_j - c^a_{ij}$

8 remove link ij: $a_{ij} \leftarrow 0, c^a_{ij} \leftarrow 0, c^b_{ij} \leftarrow 0$

9 add link ij': $a_{ij'} \leftarrow 1, c^a_{ij'} \leftarrow C^a_{ij'}, c^b_{ij'} \leftarrow C^b_{ij'}$

10 update the reserved bandwidth at the P-BS of TP i: $c^C_{j^p} \leftarrow c^C_{j^p} + max\left(\{c^a_{ij^p}\} \cup \{c^b_{ij'} \forall j' \in CS^o\}\right)$

11 update the total reserved bandwidth of BS j': $c^C_{j'} \leftarrow c^C_{j'} + c^a_{ij'} + c^b_{ij'}$

12 recalculate $z_{ijj'}$ for the improved TP i, similar to line 1

5.5 Remove obsolete BSs to reduce costs

In the fifth phase, the system costs are reduced as much as possible. This is done in two steps, first reducing the number of A-BSs and then the number of W-BSs. Both steps are explained below.

In the first step, the algorithm tries to change as much A-BSs to W-BSs as possible, as presented in *Algorithm 1.5a*. When an A-BS is changed to a W-BS, all decision variables may be changed. Of course, the primary links in p are changed, and if the primary link was not assigned as secondary link, a is changed as well. Furthermore, the backhaul links in b are also altered, and the fiber backhaul link in f is removed. If the involved FAP has no connected fiber links left, u is changed as well.

The input needed for this step includes all sets of system components. Furthermore, all decision variables are needed in this step, since they can all be altered. Additionally, the parameters denoting which links are possible (A and B) and the corresponding full demand capacities (C^a and C^b) are used as input. Finally, the parameters L and Θ are needed to calculate the AD&LL score, which is weighed by χ.

This step checks for each A-BS if all links can be moved to another A-BS. For this, a copy is made of all relevant decision variables at line 1. In these copies, the algorithm tries if all links can be moved to other A-BSs, for which the pseudo-code is given at lines 4-20. There are two conditions to be met before all links can be moved. First, each TP must be within LoS of an alternative primary A-BS, according to A. Secondly, the alternative A-BSs must have all BSs serving their TPs within LoS, according to B. If these conditions are met for all TPs is checked by moving all primary links one-by-one to another A-BS. If moving one of the links is impossible, the algorithm will not check the remaining links of the current A-BS j and continue with the next A-BS (line 17). Otherwise, if all links can be moved (lines 9-14), the changes made to the copied decision variables are applied to the real decision variables and the A-BS is changed to a W-BS at lines 21-23. Finally, Algorithm 1.4 is executed to optimize the AD&LL score before the next and final step is executed.

The complexity of this step is determined by the final score optimization, namely $O(K^2 \times n_{TP}^2)$. The rest of this step has a similar complexity when K is not considered, namely $O(n_{TP}^2)$. This is true since only primary links are considered for each A-BS. We remind the reader that the number of A-BSs is assumed to be proportional to but less than n_{TP}.

Algorithm 1.5a: Decrease the number of A-BSs

Input : TP, CS, FAP, a, p, b, f, o, w, u, A, B, C^C, C^a, C^b, L, Θ, χ

Output: a, p, b, f, and their derivatives o, w and u

let a', p', b', c'^a and c'^b be copies of a, p, b, c^a and c^b

let s^w be a flag variable indicating if the A-BS can be made wireless: $s^w \leftarrow 1$

let CS^w be the set of CSs occupied by an A-BS: $CS^w \leftarrow \{j \in CS^o | \, w_j = 1\}$

for *all A-BSs:* $j \in CS^w$*, ordered by number of connected TPs ascending* **do**

  for *all TPs with BS* j *as primary BS:* $i \in TP : p_{ij} = 1$ **do**

    let o^j be all other serving BSs of TP i: $o^j \leftarrow \{j'' \in CS^o \setminus \{j\} | \, a_{ij''} = 1\}$

    let o^p be all alternative A-BSs with TP i and all its serving BSs in LoS: $o^p \leftarrow \{j' \in CS^w | \, A_{ij'} = 1 \wedge B_{j'j''} = 1 \, \forall j'' \in o^j \setminus \{j'\}\}$

    only keep BSs in o^p that have sufficient bandwidth for all links in a & b

    if *there is an A-BS* j' *already serving TP* i*:* $\{j' \in o^p | \, a_{ij'} = 1\} \neq \varnothing$ **then**

      let j' be the first CS in o^p: $j \leftarrow o^p_0$

    else if *there is an A-BS* j' *not yet serving TP* i*:* $o^j \neq \varnothing$ **then**

      let z^p be the AD&LL-scores of the links from TP i to all A-BSs in o^p

      let j' be the A-BS in o^p with the highest score in z^p

      move the link from A-BS j to A-BS j': $a'_{ij} \leftarrow 0$, $a'_{ij'} \leftarrow 1$

    else

      removal of fiber link is impossible: $s^w \leftarrow 0$

      break: do not check other TPs of this A-BS in vain

    make the link from A-BS j' primary for TP i: $p'_{ij} \leftarrow 0$, $p'_{ij'} \leftarrow 1$

    move all backhaul links to A-BS j': $b'_{jj''} \leftarrow 0, b'_{j'j''} \leftarrow 1 \, \forall j'' \in o^j$

    update all c'-variables: $c'^a \leftarrow a' \cdot C^a$, $c'^b \leftarrow b' \cdot C^b$

  if *A-BS can be made wireless:* $s^w = 1$ **then**

    make A-BS j a W-BS: $w_j \leftarrow 0$

    replace a, p, c^a and c^b with the values of a', p', c'^a and c'^b

run Algorithm 1.4: improve AD&LL-score of all TPs

In the second step, the algorithm tries to remove as much W-BSs as possible, as presented in *Algorithm 1.5b*. This step results in the removal of BSs as from o, and the respective changes in a and b. The needed input for this consists of the CS and TP system component sets and the already formed links between them as recorded in a, p, and b. Furthermore, LoS information from A and B is needed to find alternatives for links of BSs that will be removed. Finally, the the capacity parameters and AD&LL-score calculation parameters are needed as inputs.

Algorithm 1.5b: Decrease the number of W-BSs

Input : TP, CS, a, p, b, o, A, B, C^C, C^a, C^b, L, Θ, χ
Output: a, b, and their derivative o

let a', b', c'^a and c'^b be copies of a, b, c^a and c^b
let s^r be the flag indicating if the W-BS can be removed: $s^r \leftarrow 1$
let $CS^{\overline{w}}$ be the set of all CSs occupied by W-BSs: $CS^{\overline{w}} \leftarrow \{j \in CS^o | \, w_j = 0\}$
for *all W-BSs: $j \in CS^{\overline{w}}$, ordered by number of connected TPs ascending* **do**
  for *all TPs connected to BS j: $i \in TP : a_{ij} = 1$* **do**
    let j^p be the primary BS of TP i: $p_{ij^p} = 1$
    let o^w be all unconnected W-BSs within LoS of both TP i and BS j^p:
      $o^w \leftarrow \{j' \in CS^o | \, a_{ij'} = 0 \wedge A_{ij'} = 1 \wedge B_{j'j^p} = 1\}$
    only keep BSs in o^w that have sufficient bandwidth for all links in a & b
    if *there is a BS available in o^w: $o^w \neq \varnothing$* **then**
      let z^w be AD&LL-scores of the links from TP i to all BSs in o^w
      let j' be the BS in o^w with the highest score in z^w
      move the link from W-BS j to W-BS j': $a'_{ij} \leftarrow 0$, $a'_{ij'} \leftarrow 1$
      move the backhaul from W-BS j to W-BS j': $b'_{jj^p} \leftarrow 0$, $b'_{j'j^p} \leftarrow 1$
      update all c'-variables: $c'^a \leftarrow a' \cdot C^a$, $c'^b \leftarrow b' \cdot C^b$
    else
      conclude removal is impossible: $s^r \leftarrow 0$
      break: do not check other TPs of this BS in vain
  if s^r **then**
    remove W-BS j itself: $o_j \leftarrow 0$
    remove j from CS^o, add it to CS^u: $CS^o \leftarrow CS^o \setminus \{j\}$,
      $CS^u \leftarrow CS^u \cup \{j\}$
    replace a, p, c^a and c^b with the values of a', p', c'^a and c'^b
run Algorithm 1.4: improve AD&LL-score of all TPs

This step works very similar to the first step of this phase. The involved decision variables are duplicated in x' variables at line 1. It tries to move all links of a W-BS to other BSs at lines 5-17. It must be possible to move both access and backhaul links, as checked in lines 6-9. If this proves impossible, the loop will break and the BS will not be removed (line 17). After all link of a W-BS are removed, it is removed and the resulting links are copied to the decision variables at lines 18-21. Finally, Algorithm 1.4 is executed to optimize the AD&LL score before the algorithm finishes.

The complexity of this step is again determined by the final score optimization, namely $O(K^2 \times n_{TP}^2)$. The same reasoning as for the previous step holds for this step. The overall complexity of this step is therefore $O(K^2 \times n_{TP}^2)$.

Chapter 6

Algorithm performance

In this chapter, the performance of the MIND-GO algorithm is analyzed. This is expressed in terms of the objective functions as defined in Chapter 4: deployment costs and blockage sensitivity. We have conducted four test rounds, as shown in Figure 6.1.

In the first test round, the weight parameters Γ^A and Γ^W for scoring candidate sites are investigated. These parameters are used to determine the score of each candidate site (CS) from its partial scores, as described in Chapter 5. These scores determine where A-BSs and W-BSs are deployed. Based on the outcomes, the parameter settings for subsequent test rounds are chosen, such that mainly the deployment costs will be minimized. If it is possible to decrease the blockage sensitivity without significantly increasing the costs, that weight parameter setting is preferred.

In the second test round, we investigate the improvement achieved in each of the five phases of the algorithm. This is mainly interesting for the final two phases, since the fourth phase optimizes the AD&LL score (see Section 5.4), whereas the final phase removes as much BSs from the system as possible (see Section 5.5). Removing BSs results in a lower AD&LL score, of course, so it is good to know how far this score is reduced.

In the third test round, the algorithm's results are compared to the brute force optimal solutions. This provides insights in how well the heuristic algorithm approaches the optimal score with its used budget.

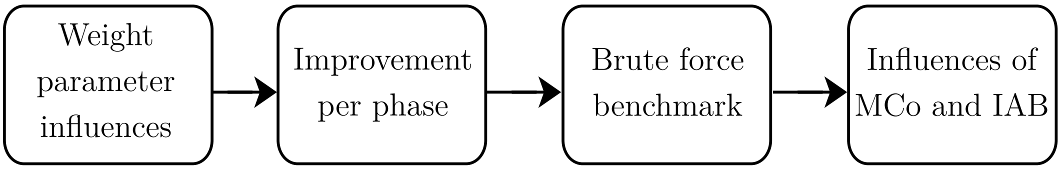

Figure 6.1: Test rounds to gain insight in the MIND-GO algorithm

In the final test round, we determine the influence of MCo and IAB on the network deployment. The results are analyzed from two perspectives, according to our subquestions as stated in Chapter 1. Firstly, the influence on the system costs and performance are discussed. For this, the value of K is varied to find out how this influences the AD&LL score. Secondly, the influence on the needed bandwidth for IAB is discussed. For this, network deployments using IAB are compared to those consisting of only fiber backhauled BSs.

Before these test rounds are presented in subsequent sections, the next section presents the chosen parameter settings for our simulation environment. Next, the four subsequent sections describe the performance analysis of the studied scenarios, as described above. Finally, we summarize the results in Section 6.6.

6.1 Simulation environment setup

We have developed a system-level simulator in MATLAB to analyze the performance of our heuristic algorithm. This simulator is able to randomly generate maps with TPs and FAPs on them and provides the map data in the required parameters to the algorithm. This is described in two steps below. Then, the parameter settings of the channel model are presented. Finally, we describe all input parameters required by the MIND-GO algorithm.

Map layout generation

Table 6.1 lists the parameters that must be provided for the map layout generation. This results in buildings in a regular grid, as shown in Figure 6.2. The first five parameters are used to generate the map layout, and the next four parameters define how TPs and FAPs are placed on the map. Although the number of CSs on the maps is strictly no parameter, for the sake of clarity their values are listed in Table 6.1 for two map sizes (default and benchmark).

As the city environment, we will consider a regular grid map of blocks sized 100×50 m. The streets are 20m wide, including sidewalks of 3 m on each side. The CSs are located on the corner of each block, and in the middle of each side of a block. This setup is similar to that used by Palizban et al. in [10]. Only at the edges of the map, no CSs will be present. This results in a maximum distance between CSs of 50 m, which is half the width of a block. The total number of CSs on the two map sizes is $n_{CS} = 136$ for default maps and $n_{CS} = 22$ for benchmark maps.

Table 6.1: Map parameters and their values for the default and benchmark settings

Parameter	Notation	Default	Benchmark
block dimensions		100×50 m	
road width		14 m	
sidewalk width		3 m	
map dimensions		4×6 (460×400 m)	2×3 (220×190 m)
open space chance	P_{OS}	20%	0, 33, 100%
number of TPs	n_{TP}	10 - 80	5
HPPP TP density	λ_{TP}	$n_{TP}/400$	0.0125
number of FAPs	n_{FAP}	20	5
HPPP FAP density	λ_{FAP}	0.1	0.025
number of CSs	n_{CS}	136	22

Additionally, to make the map more realistic, a portion of all blocks will have no building on it. This is referred to as an open space. The percentage of blocks that will be left open is set by the parameter P_{OS}. CSs will be present at the same locations around open spaces as for buildings.

Placement of TPs and FAPs

Next, TPs and FAPs are placed using a homogeneous poisson point process (HPPP) as in [44]. TPs are placed on the roads (excluding sidewalks), as they are assumed to be vehicles, whereas FAPs are placed on the sidewalks. Both are never placed outside the outer CSs. This is called the area of interest (AoI). Their locations are generated as described below. We will explain this for TPs, the process for FAPs proceeds analogously. The locations are first determined using a HPPP distribution with density λ_{TP} within the AoI. All locations that are not on the roads (for TPs) or sidewalks (for FAPs) are then removed. If the remaining locations exceed the desired number as set by the parameter n_{TP}, only the first n_{TP} TPs are kept. In case the remaining locations are less than n_{TP}, new locations are generated as described. An example of a generated 4×6 map is shown in Figure 6.2. This map size is used in most simulations, except for the simulations in which the brute force algorithm is used. In order to keep the running time within an acceptable limit of several hours, a map of only 2×3 blocks is used in those cases. This is referred to as a benchmark map. The default settings for both map variants are listed in Table 6.1.

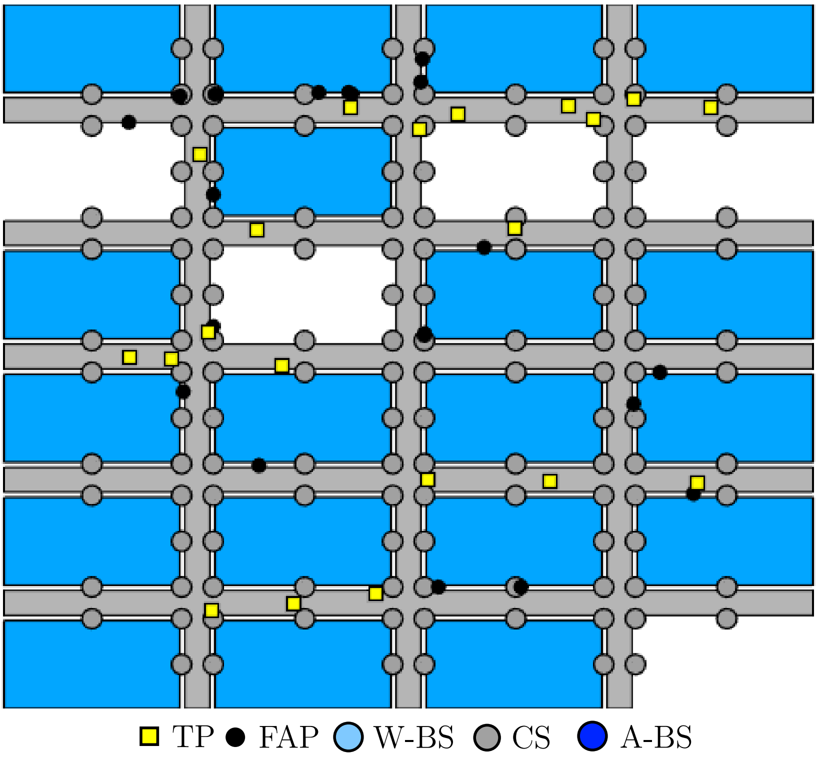

Figure 6.2: Default map with 20 TPs, 20 FAPs, 136 CSs, and a 20% open space chance

Channel model parameters

The values that we used for the channel model are listed in Table 6.2. We assumed the transmit power to be 30 dBm, as used by both [8] and [37]. The noise is calculated as $-174 + log_{10}(BW)$ like in [9], where the bandwidth (BW) is set to 800 MHz (like C^C, see Table 6.3). This results in $-174 + 89 = -85$ dBm. For the path loss formula, we used the values from the work of Sulyman et al. [26], who based these on empirical data from New York and Manhattan. They found the values for the path loss formula to be $\alpha = 0.9$ and $\beta = 1.8$ for LoS communication at 28 GHz, using antennas with a beam width of 28.8 °and 15 dBi gain for both G^{TX} and G^{RX}.

Table 6.2: Parameters and values for the channel model

Parameter	Notation	Value
transmit power	P^{BS}	30 dBm
transmit gain	G^{TX}	15 dBi
receive gain	G^{RX}	15 dBi
noise power	P^N	-85 dBm
path loss exponent	α	0.9
slope correction factor	β	1.8

Table 6.3: Parameters for the MIND-GO algorithm and their values for the default and benchmark settings

Parameter	Notation	Value
BS reach	L^{max}	150 m
demand of each TP	D	1000 Mbps
bandwidth capacity of each BS	C^C	800 MHz
coverage value for each TP	K	3
base costs of a BS	E^D	1
additional costs of an A-BS	E^A	1
costs of using an FAP	E^F	0.05
costs of fiber cable deployment per meter	E^C	0.02
weight for AD&LL scores	χ	$\{0.5, 0.5\}$

Input parameters for the algorithm

Two types of input parameters are set for the algorithm. The first group of these input parameters are calculated from the scenario that is generated by the simulator as described above. The second group of parameters define the constraints on the network deployment. Both are described below. All parameters that cannot be derived from the generated scenario are listed in Table 6.3.

The matrices that define the scenario as input for the algorithm are filled with distances, LoS link possibility, and link angles. The relevant distances between any two system components are calculated. L_{ij} stores the distance between TPs and CSs, $S_{jj'}$ stores the distance between each pair of CSs, and F_{jp} stores the distance between CSs and FAPs. The values of A_{ij} and $B_{jj'}$ are then determined, based on two factors: (1) if the distance between the two points is less than the BS reach; and (2) if no building is located directly between both points. The BS reach is set to 150 m (Table 6.3), since this is the distance reported in multiple studies at which mmWave communication is feasible [41], [45]. The link angle matrix $\Theta_{ijj'}$ is filled with the angles between any pair of links that connect two BSs j, j' to the same TP i. These parameters are explained in more detail in Chapter 3.

For all possible links, the needed bandwidth for serving TPs their full demand is calculated using the channel model. This is done for both access (C^a) and backhaul links (C^b), as described in Section 3.2. The demand of each TP (D) is set to 1000 Mbps, and the available bandwidth capacity of each BS (C^C) is set to 800 MHz (based on [9]).

Beside the map data, eight other parameter values are required by the MIND-GO algorithm. These are all listed in Table 6.3. The coverage value parameter K defines the number of links that each TP must be provided with, as explained in Section 3.1. Then, five parameters are required to determine the values of the objective functions as described in Chapter 4. The primary objective minimizes the deployment costs. For this, four parameters named E^x are needed, as explained in Section 4.2. The values for these parameters come from the work of Rezaabad et al. [9]. They have abstracted these values from more precise values of multiple other studies. The secondary objective minimizes the blockage sensitivity, defined as a combination of angular diversity (AD) and link length (LL) in Section 4.3. The weights for these two values must be given by the parameter χ, which is set to weigh them equally. This value is based on the research outcomes of Devoti et al. [8].

In the following sections, time measurements are included to provide insight in the time-complexity of the MIND-GO algorithm. To put these numbers in perspective, it might be helpful to know that the simulations are performed on a 2021 Surface Laptop 4 with an Intel® Core™ i7-1185G7 @ 3.00 GHz (5.00 GHz boost) and 16 GB of LPDDR4x RAM @ 4267 MHz.

6.2 Impact of parameter settings for CS scoring

Now, we will first evaluate the impact of two parameter settings for the MIND-GO algorithm that define the weights for partial scores for CSs. These have been introduced in Sections 5.1 and 5.3. The first parameter is for the scoring of CSs for A-BS deployment (Γ^A), and the other is for the scoring of CSs for W-BS deployment (Γ^W). Both parameters are briefly explained below.

When scoring CSs for A-BS deployment, three properties of each CS are considered, namely (1) the length of the fiber link to the closest FAP, (2) the number of TPs within LoS that are not fully served their demand yet, and (3) the average AD&LL score for all TPs that are within LoS (including already covered ones). The first two are aimed at reducing the system costs, while the last one accounts for the system performance under dynamic blockages. When scoring CSs for W-BS deployment, only the last two factors are relevant. Below, we explore how different values for these parameters influence the performance and costs of the network deployed by the MIND-GO algorithm.

6.2.1 Exploring weight parameter values for A-BS Deployment

To investigate the influence of the weight parameters for A-BS deployment on the deployment costs and network performance, two sets of tests are run. First, we look into the balance between the weights for costs and the weight for system performance (AD&LL). Based on the outcomes, we choose a value for the system performance weight. Then, we further investigate the balance between the two weights aimed at decreasing the deployment costs.

Setup of tests for costs versus performance balance

First, we investigate the influence of the balance between the weights for costs and the weight for system performance (AD&LL). For this, tests are run with six parameter settings, referred to as scenarios 1-6 (see Table 6.4). In these scenarios, the weight gradually shifts in steps of 0.2 from the final parameter weight (for AD&LL score) towards the other two parameters (for costs minimization). The weight is equally distributed among the costs weight parameters. The value of Γ^W in this simulation is set to $\{0.5, 0.5\}$ in each scenario.

To see if the results are dependent on the number of TPs, we also vary n_{TP} from 10 to 80 with steps of 10. We report the average of 1000 runs for each scenario along with a 95% confidence interval. For each run, one default map is randomly generated to be used in all scenarios (see Table 6.1).

Table 6.4: Shifting scoring weight parameters from AD&LL to costs for A-BS deployment

Scenario	Γ^A	Γ^W
1	$\{0.0, 0.0, 1.0\}$	$\{0.5, 0.5\}$
2	$\{0.1, 0.1, 0.8\}$	$\{0.5, 0.5\}$
3	$\{0.2, 0.2, 0.6\}$	$\{0.5, 0.5\}$
4	$\{0.3, 0.3, 0.4\}$	$\{0.5, 0.5\}$
5	$\{0.4, 0.4, 0.2\}$	$\{0.5, 0.5\}$
6	$\{0.5, 0.5, 0.0\}$	$\{0.5, 0.5\}$

Results of tests for costs versus performance balance

It appears from Figure 6.3 that scenario 6 result in the lowest system cost. In Table 6.4 it can be seen that in this scenario no weight is put on the third score aspect, which is for AD&LL. Therefore, we would expect to see a lower system score (AD&LL) for this scenario in Figure 6.4 than for the other scenarios. However, this is only the case for the lower n_{TP} values. Even then, the decrease in score is not significant when considering the large confidence interval. This might be attributed to the fact that the AD-part of the score is not influenced by the deployment of A-BSs. The deployment of W-BSs will probably have more impact, and the optimization phases of our algorithm. Since we set reducing the costs as our primary goal, we will select scenario 6 as the best option. Therefore, the last value in Γ^A will be set to 0 in all next simulations.

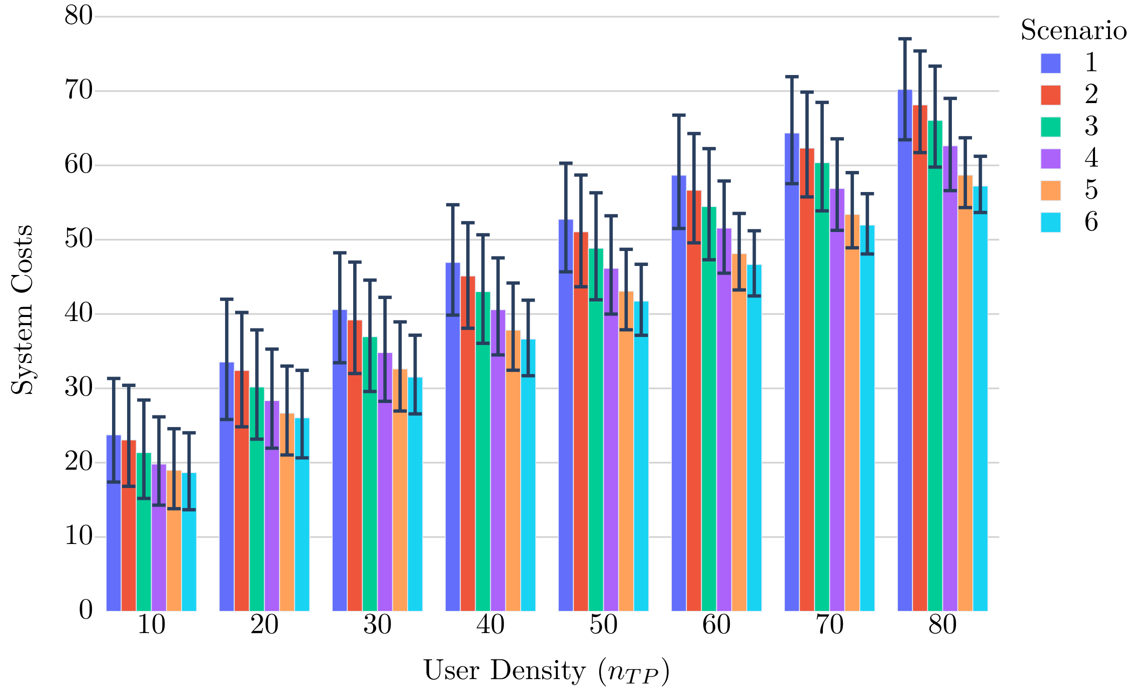

Figure 6.3: System costs for all six scenarios from Table 6.4

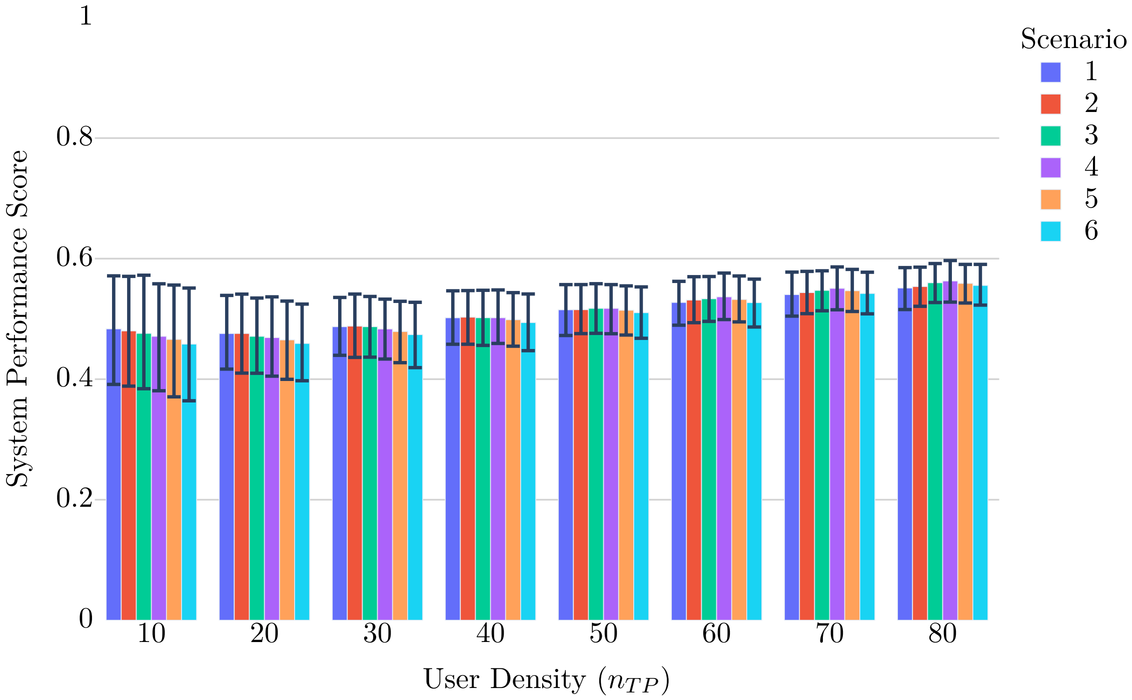

Figure 6.4: AD&LL scores for all six scenarios from Table 6.4

Setup of tests for costs weights balance

Next, we investigate the influence of the scoring weights that aim to decrease the deployment costs. The first of these is the length of the fiber backhaul that is needed when an A-BS is placed on a CS, the other is how many TPs it will cover. For this, we run tests with seven scenarios as listed in Table 6.5. In these scenarios, the weight gradually shifts in steps of 0.2 from full weight on fiber length to full weight on TPs covered. The scenario in which the weight is distributed equally amongst them is also included (scenario 4). The value of Γ^W in this simulation is set to $\{0.5, 0.5\}$ in each scenario.

To see if the results are dependent on the number of TPs, we also vary n_{TP} from 10 to 80 with steps of 10. We report the average of 1000 runs for each scenario along with a 95% confidence interval. For each run, one default map is randomly generated to be used in all scenarios (see Table 6.1).

Table 6.5: Shifting scoring weight parameters for costs from fiber link length to number of covered TPs for A-BS deployment

Scenario	Γ^A	Γ^W
1	$\{1.0, 0.0, 0.0\}$	$\{0.5, 0.5\}$
2	$\{0.8, 0.2, 0.0\}$	$\{0.5, 0.5\}$
3	$\{0.6, 0.4, 0.0\}$	$\{0.5, 0.5\}$
4	$\{0.5, 0.5, 0.0\}$	$\{0.5, 0.5\}$
5	$\{0.4, 0.6, 0.0\}$	$\{0.5, 0.5\}$
6	$\{0.2, 0.8, 0.0\}$	$\{0.5, 0.5\}$
7	$\{0.0, 1.0, 0.0\}$	$\{0.5, 0.5\}$

Results of tests for costs weights balance

It appears from Figure 6.5 that not one scenario results in the lowest system cost for all user densities. Only the seventh scenario results in higher deployment costs. In this scenario, the fiber link length was not taken into account. This clearly results in extra costs, especially for higher user densities. Since we have decided to put no weight on the AD&LL score weight for A-BS deployment, it was expected to see no difference in the system scores among the seven scenarios. From Figure 6.6 we conclude that this holds, although a slight and insignificant influence on it can be observed for lower user densities. Although there is no apparent winner, we will select scenario 4 as parameter weight for A-BS deployment. Therefore, the value of Γ^A will be set to $\{0.5, 0.5, 0.0\}$ in all next simulations.

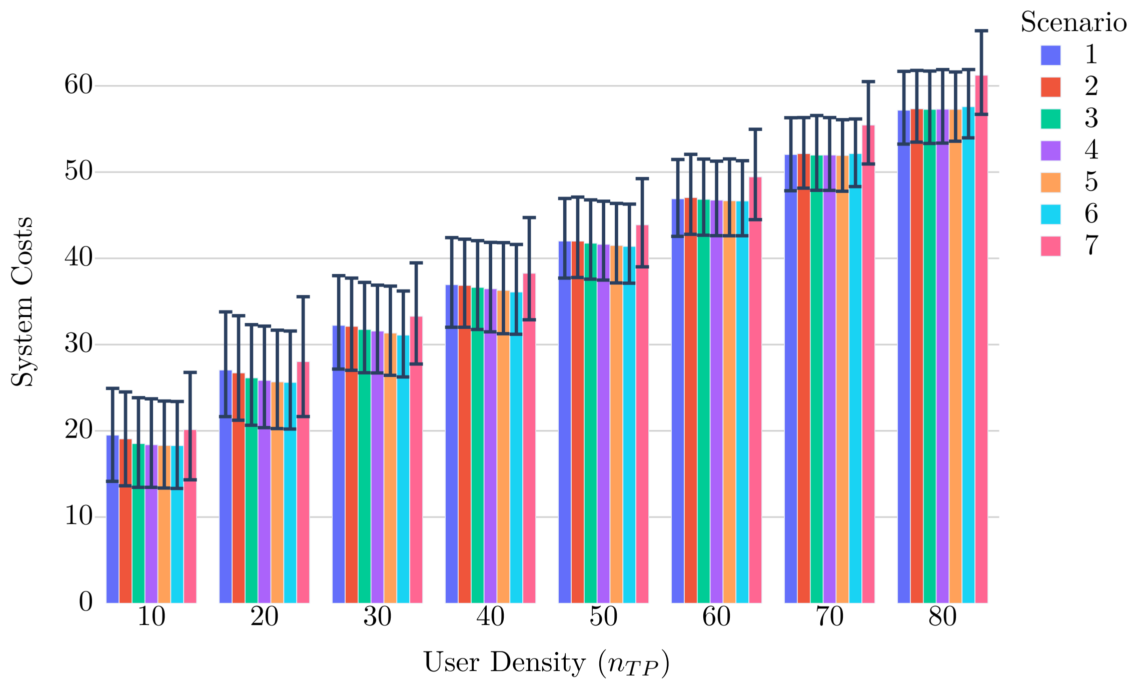

Figure 6.5: System costs for all seven scenarios from Table 6.5

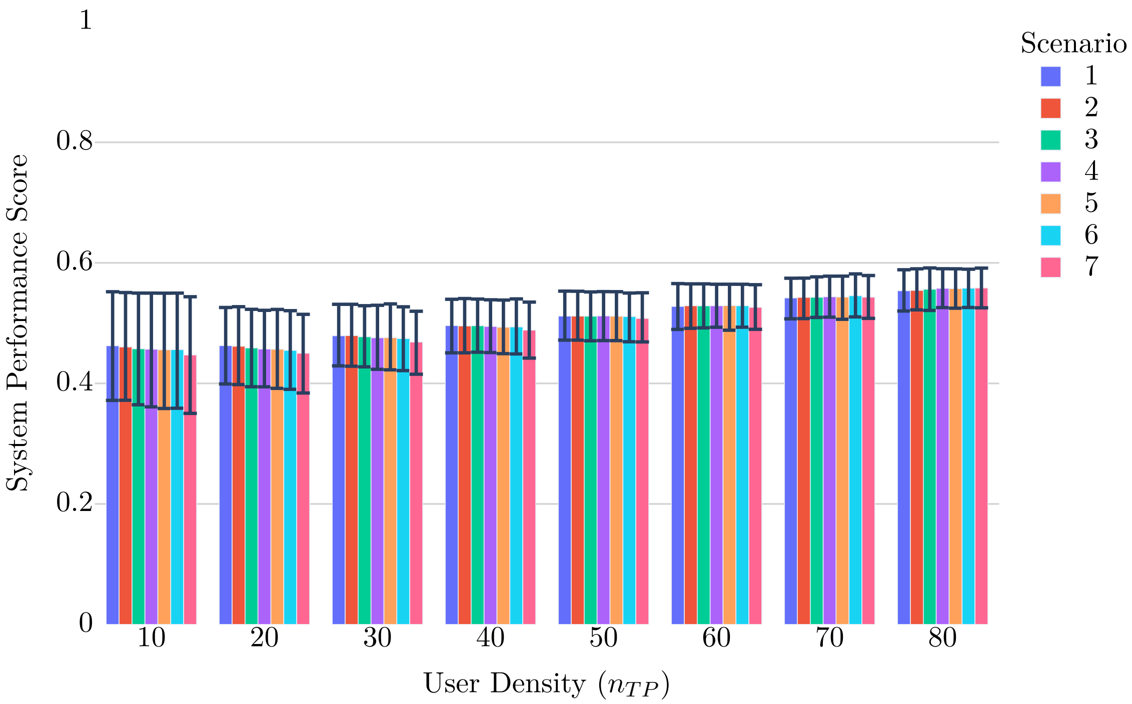

Figure 6.6: AD&LL scores for all seven scenarios from Table 6.5

6.2.2 Exploring weight parameter values for W-BS Deployment

Setup

We further investigated the influence of setting Γ^W to different values. For this, we defined seven scenarios, as listed in Table 6.6. In these scenarios, the weight is shifted in steps of 0.2 from covered TPs to AD&LL score. Again, we added scenario 4 in which the weights are equally distributed among these two partial scores.

To see if the results are dependent on the number of TPs, we also vary n_{TP} from 10 to 80 with steps of 10. We report the average of 1000 runs for each scenario along with a 95% confidence interval. For each run, one default map is randomly generated to be used in all scenarios (see Table 6.1).

Table 6.6: Scoring weight parameter values varying for W-BSs deployment

Scenario	Γ^A	Γ^W
1	$\{0.5, 0.5, 0.0\}$	$\{1.0, 0.0\}$
2	$\{0.5, 0.5, 0.0\}$	$\{0.8, 0.2\}$
3	$\{0.5, 0.5, 0.0\}$	$\{0.6, 0.4\}$
4	$\{0.5, 0.5, 0.0\}$	$\{0.5, 0.5\}$
5	$\{0.5, 0.5, 0.0\}$	$\{0.4, 0.6\}$
6	$\{0.5, 0.5, 0.0\}$	$\{0.2, 0.8\}$
7	$\{0.5, 0.5, 0.0\}$	$\{0.0, 1.0\}$

Results

Figure 6.7 shows that the costs increase when the weight shifts towards AD&LL. The difference between the first five scenarios is however not very significant. Moreover, in Figure 6.8 is can be seen that the system performance increases simultaneously. Therefore, we have chosen scenario 4 and the value of Γ^W is set to $\{0.5, 0.5\}$ in the remaining tests.

Discussion

Thus far, for the higher user densities, the system score seems to be mainly determined by other steps in the algorithm than the initial placement of A-BSs and W-BSs. We will explore this issue further in Subsection 6.3.

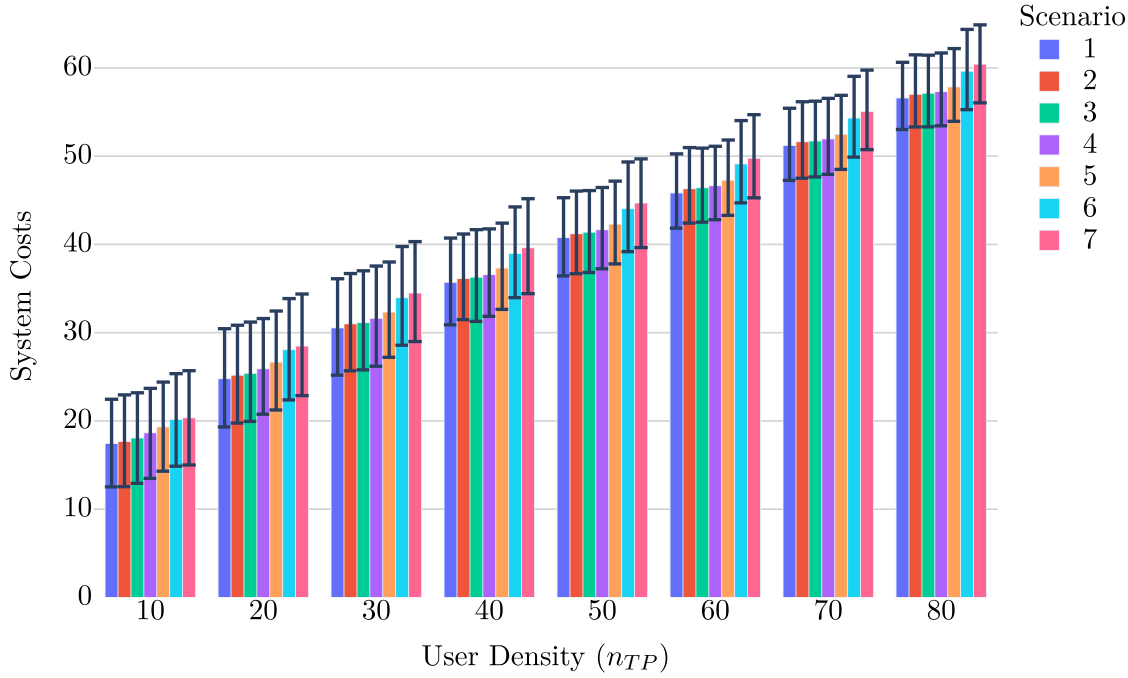

Figure 6.7: System costs for all seven scenarios from Table 6.6

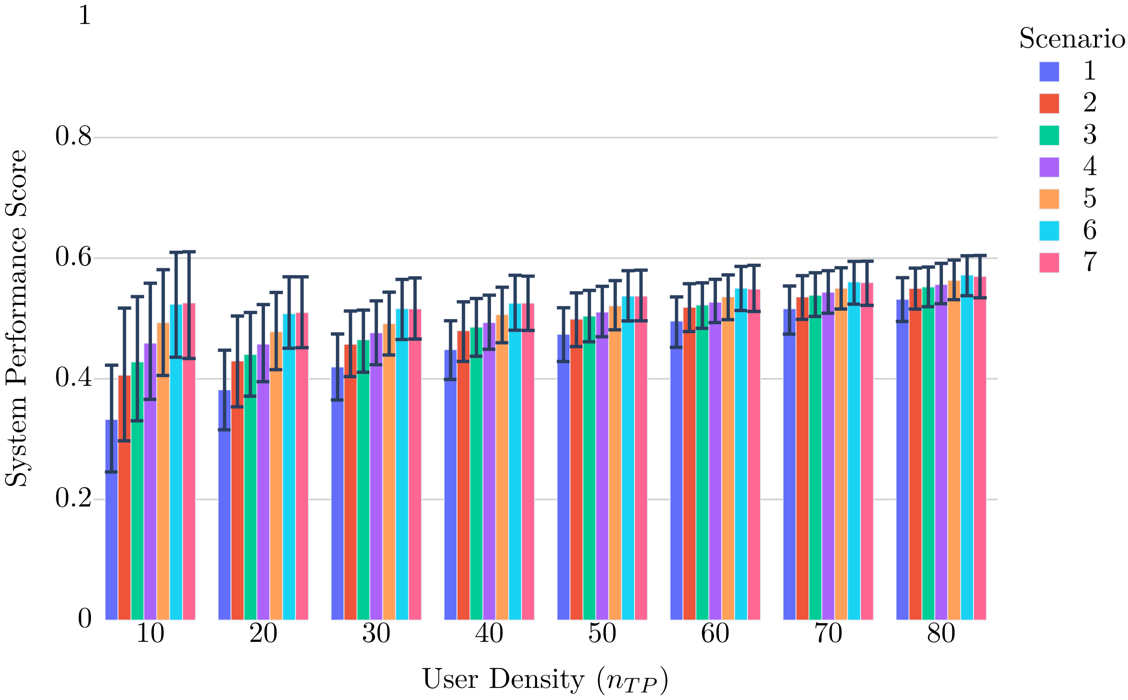

Figure 6.8: AD&LL scores for all seven scenarios from Table 6.6

6.3 Impact of each algorithm phase

As described earlier in Chapter 5, the algorithm runs in five phases: (1) A-BS deployment, (2) A-BS optimization, (3) W-BS deployment, (4) AD&LL optimization, and (5) BS removal. In this section, the effects of these phases on system score and costs are presented, as well as the time these take for the MIND-GO algorithm to run. A total of 1000 default maps are generated, on which measurements are done after each phase. We report the average of these runs for each phase along with a 95% confidence interval.

We will discuss the impact of each phase based on Figure 6.10 and 6.9 that depict the number of deployed BSs and the system score, respectively. The number of deployed BSs makes up most of the system costs. Therefore, no additional graph is included showing the system costs. Before we start analyzing, it is good to note that no AD score can be calculated in phases 1 and 2, because each TP is served by exactly 1 A-BS in those phases. Hence, the AD score is set to zero. This results in the lower AD&LL score observed in Figure 6.9 for phases 1-2. Now, we will analyze the impact of each phase on the AD&LL score and the number of deployed BSs.

Phase 1: Initial deployment of A-BSs for 1-connectivity
Figure 6.10 shows that the number of A-BSs after phase 1 is significantly higher than after the second phase. This is caused by the way we initially deploy A-BSs, namely by placing it at a CS and then equally dividing its bandwidth over all TPs within LoS. This is repeated until all TPs can be served their full demand, but then they are served by multiple A-BSs. Hence, the links are reorganized at the end of phase 1. This results in a high number of A-BSs that each serve only a few TPs.

Phase 2: Remove A-BSs and select primary links
In the second phase all remaining bandwidth is used to assign as much links as possible, starting with the shortest links. A-BSs of which all links have then become redundant are removed. This results in a lower system cost (Figure 6.10), but also in a slight decrease of the LL score (Figure 6.9), since all TPs were connected using the shortest possible links to the deployed BSs at the end of phase 1. Then, after selecting the shortest link as primary link for each TP, any other link is removed.

Phase 3: Initial deployment of W-BSs for K-connectivity In the third phase, the AD&LL scores are increased by providing the TPs with K-connectivity ($K = 3$). The number of W-BSs deployed for this appears to be linearly increasing with the user density n_{TP} (Figure 6.10). Nevertheless, the improvement of the AD&LL score is less for higher user densities than for lower user densities (Figure 6.9). This might very

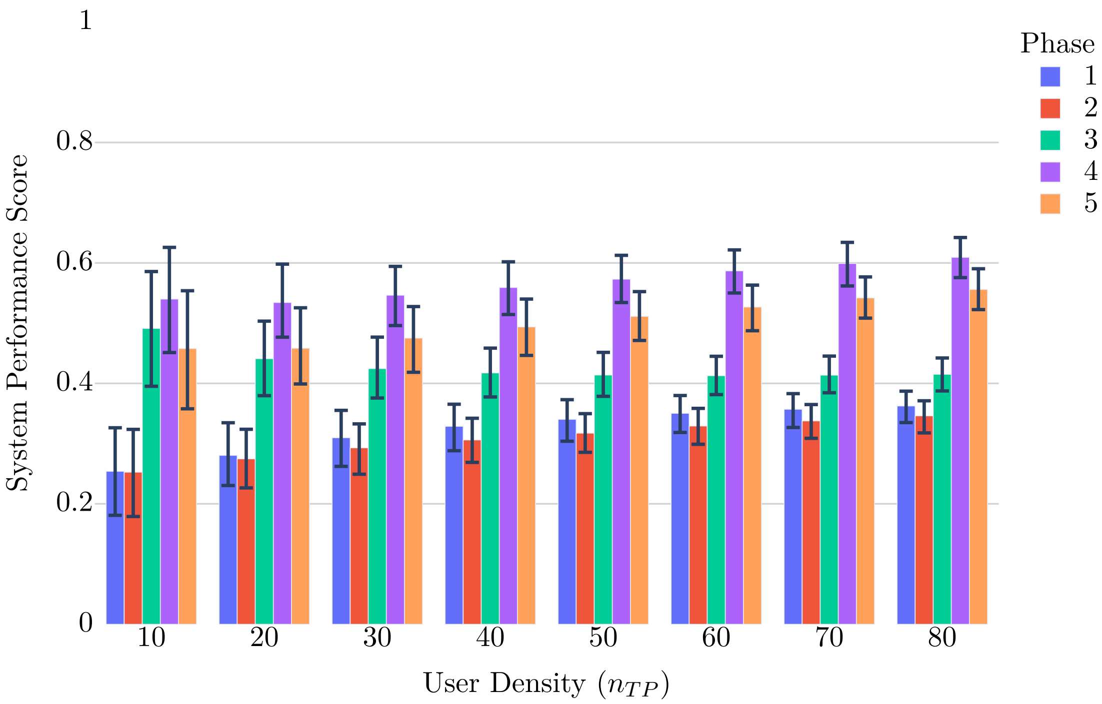

Figure 6.9: AD&LL score per phase

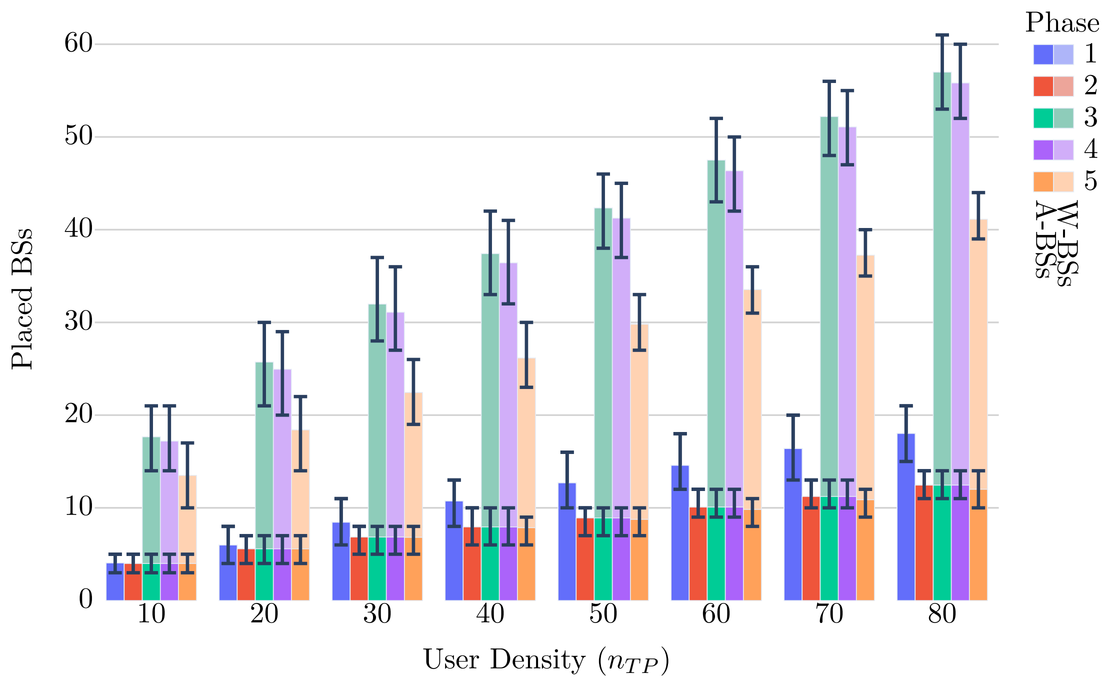

Figure 6.10: Placed BSs per phase

well be due to the greedy approach that we use to select links. Links with the highest resulting score are namely selected when a W-BS is placed. This clearly does not lead to an optimal link assignment, as the next phase manages to increase the AD&LL score significantly.

Phase 4: Swap links to optimize AD&LL-score

In the fourth phase, the system score is observed to be enhanced again in Figure 6.9. This is achieved by swapping secondary links from TPs to other deployed BSs as long as this improves the system score. Although score improvement is the main goal of this phase, it may result in W-BSs with no links left. These are removed at the end of the phase, which results in a slight decrease of placed BSs, as can be seen in Figure 6.10.

Phase 5: Remove obsolete BSs to reduce costs

In the final phase, as much BSs as possible are removed by swapping links of BSs having few links to other BSs. For each swap, the best alternative link is chosen, but this still inevitably results in a decreased system score (Figure 6.9). However, this decrease is small (14.8% for 10 TPs - 8.2% for 80 TPs) compared to the number of BSs that are removed (22% for 10 TPs - 26% for 80 TPs), especially on a map with many TPs, see Figure 6.10.

Running times

Finally, we discuss the running times as presented by Figure 6.11. The running times of the first three phases are very low compared to phase 4 and 5. The former take time in the order of tenths of seconds, whereas the latter exhibit a polynomial growth to some seconds. The reason for this is presumably that all links from a TP are compared to other unused links in these phases.

The complexity of each phase was expected to be in $O(n_{TP}^2)$, as described in Chapter 5. When we analyzed the results, however, the second phase could be fitted to a linear curve with $R^2 = 0.9939$, whereas the fourth and fifth phases seem to be in $O(n_{TP}^3)$ with $R^2 > 0.99$. Only the first and third phase could be fitted to a second order polynomial with $R^2 > 0.99$. Fitting the running times of phase 4 and 5 to a second order polynomial resulted in $R^2 > 0.97$, but the slope did not fit for the highest values. The reason for this is probably that comparing used links to unused links results in a complexity of $O(n_{TP} \times n_{BS}^2)$, whereas we expected the number of unused links to be in $O(K \times n_{BS})$ instead of $O(n_{BS}^2)$. Since we have observed that the number of BSs (n_{BS}) is closely related to (n_{TP}), $O(n_{TP} \times n_{BS}^2)$ can be translated to a complexity of $O(n_{TP}^3)$.

This raises the question if the phases 4 and 5 should be skipped. We argue that this should not be done, since these are the only phases that improve the system efficiency by respectively promoting the AD&LL score and decreasing the costs, which effects are especially seen for higher user densities in both Figures 6.9 and 6.10. It should also be noted that a polynomial algorithm is very efficient for network planning [10].

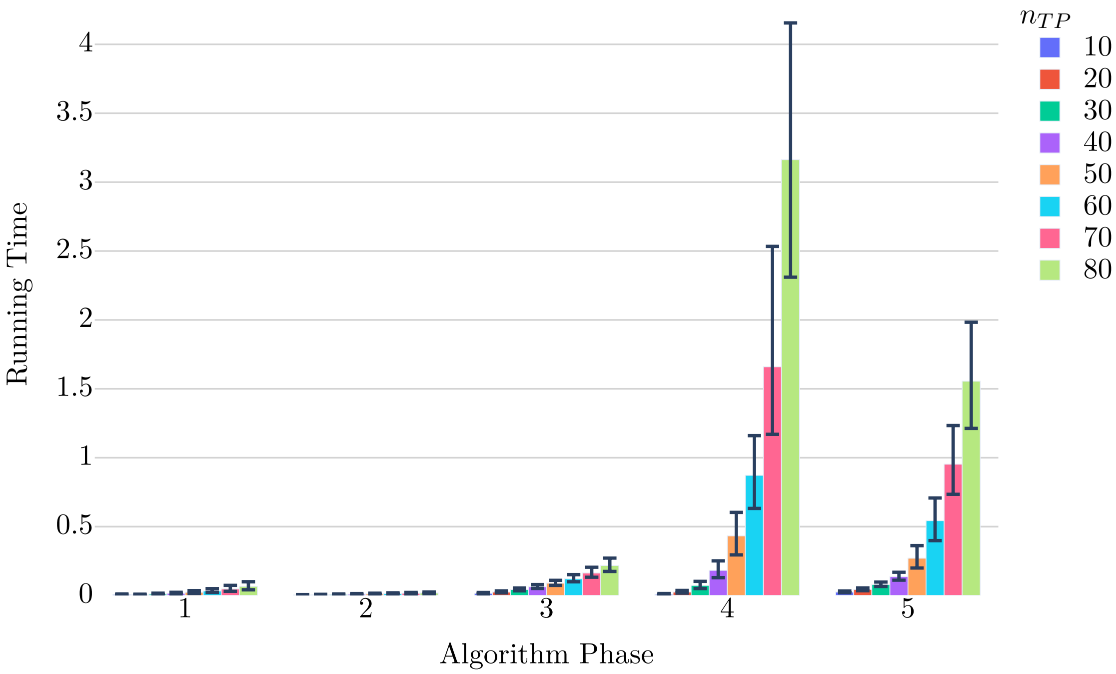

Figure 6.11: Running time of the algorithm up to each phase

6.4 Comparison with optimal and baseline algorithms

To assess the performance of the MIND-GO algorithm, the deployed network is compared with a pure greedy baseline algorithm and a brute force optimal solution. This is done on 1000 randomly generated 2×3 benchmark maps (see Table 6.1). To explore the impact of different city layouts, three variant of each map are generated. First, all buildings are present, then two of the six buildings are removed, and finally all buildings are removed. The locations of TPs and FAPs remain the same on each of these three map variants.

The brute force algorithm solves the optimization problem that is described in Chapter 4. For this, it takes the map and a budget, and determines if there are possible solutions with a lower budget. From these cheapest solutions, the one with the highest AD&LL score is selected. The brute force algorithm does this by determining the costs and the score of each possible deployment within the budget. The budget is the resulting costs of the MIND-GO algorithm.

The pure greedy baseline algorithm deploys a network by performing only phases 1 and 3 (A-BS and W-BS deployment) of the MIND-GO algorithm. This algorithm is included since the greedy approach is known to be the easiest solution to cell planning problems. This algorithm is named MIND-G, as it leaves out the optimization phases of the MIND-GO algorithm. It also does a slight modification to phase 1. The MIND-GO algorithm namely divides the bandwidth of each A-BS equally over all covered TPs in the first phase. The MIND-G algorithm, however, always reserves sufficient bandwidth for full demand links with a maximum link length (L^{max}) for its closest TPs. This way we make sure that enough bandwidth is available for wireless backhaul links.

Figure 6.12 shows that the average costs of the brute force optimum is almost the same as for the MIND-GO algorithm. When we investigated this on the raw data, it appears that the average improvement is 0.3%, 0.5%, and 0.0% for the three different open space chance values respectively. The brute force algorithm uses the same budget as the MIND-GO algorithm 89.2%, 91.4%, and 100% of the runs for the three different open space chance values respectively. In the other the runs, the budget is improved on average with 4.2% (at most 24.3%) by deploying less BS (1.7%, 4.0%, and 0.0%) or by using less fiber cable (9.1%, 4.6%, and 0.0%). These results indicate that the MIND-GO algorithm does deploy a network for a very close to optimal budget most of the time. However, the scenario is very small, so more extensive simulations must be performed to make a firm statement about this.

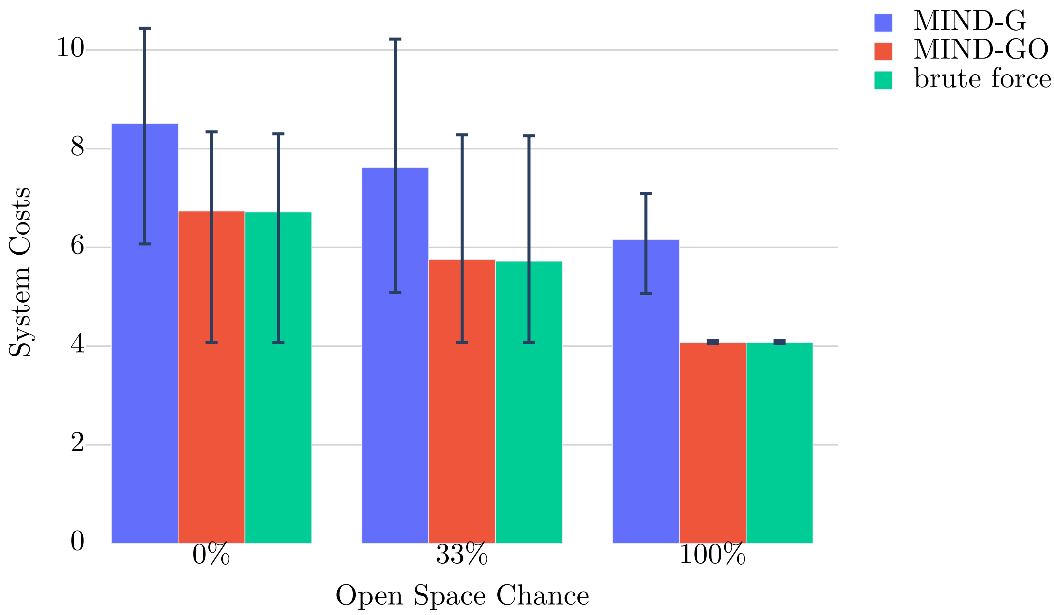

Figure 6.12: Network deployment costs for comparing the MIND-GO algorithm with a baseline solution (MIND-G) and an optimal solution (brute force)

Figure 6.13 shows that the baseline algorithm results in a higher AD&LL score than our optimized algorithm on average. It does so, however, at a much higher budget (see Figure 6.12). Furthermore, the higher AD&LL score was also expected based on Figure 6.9, since it shows a slightly higher score in phase 3 compared to phase 5 for lower user densities. This does not hold for higher user densities however, so this is mainly a result of the small scenario used in this simulation round.

It is also observed in Figure 6.13 that average AD&LL score of the MIND-GO algorithm is 95.7%/95.1%/90.2% of the brute force optimum (for open space chances 0/33/100). However, this decreases when we account for the lower deployment costs of some brute force deployments. When the deployment costs are equal, the percentages slightly drop to 94.8%/93.5%/90.2%. So, for small scenarios we see that the MIND-GO algorithm performs within 90% of the AD&LL score of the optimal solution.

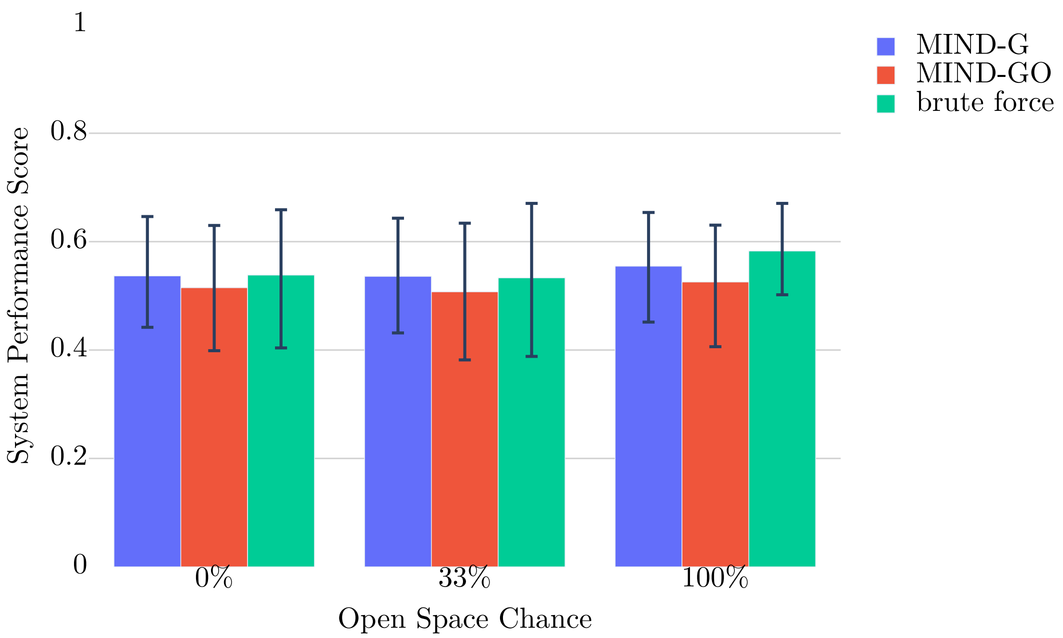

Figure 6.13: AD&LL scores for comparing the MIND-GO algorithm with a baseline solution (MIND-G) and an optimal solution (brute force)

When looking at Figure 6.14, two things are remarkable. Firstly, the angular diversity increases with the open space chance, while the link length score decreases. Regardless, the overall AD&LL score does not increase drastically. Therefore, it seems that the LL score can compensate the lower AD score at least partially. The AD and the LL values of the brute force optimum even converge when no buildings are present. This matches with the findings of Devoti et al. [8], whereas they did not include any static blockages, finding AD and LL partial scores being roughly equal. The gap between our AD and LL scores can likely be explained by the smaller number of CSs within LoS when buildings are present. It might be very well possible that enabling the algorithm to move BSs away from their starting positions (CSs), the AD score might be increased while the LL score is decreased, resulting in overall higher score.

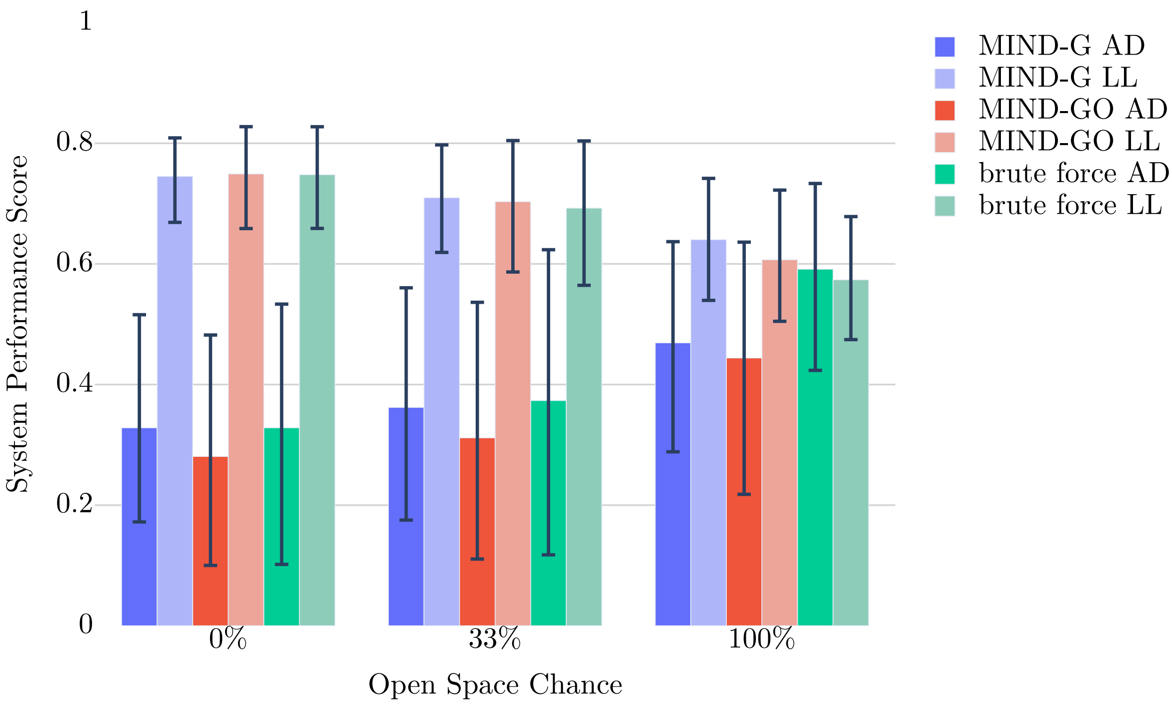

Figure 6.14: Split AD and LL scores for comparing the MIND-GO algorithm with a baseline solution (MIND-G) and an optimal solution (brute force)

6.5 Implications of Multi-Connectivity with self–backhauling

6.5.1 Implications of MCo on performance and costs

In this subsection, an answer is sought to our first sub-question:

Q1. *What are the implications of the requirement for multi-connectivity on mmWave BS placement?*

For this, 1000 default maps are generated, on which measurements are done for user densities from 10 to 80 in steps of 10, and for coverage parameter (K) values ranging from 1 to 5. We report the average values of these runs for deployment costs, AD&LL scores, and assigned bandwidths, along with a 95% confidence interval. The results are presented to show the influence of MCo on (1) the system costs and (2) the system performance, respectively.

Implications of MCo on deployment costs
First of all, the costs are predominantly determined by the number of placed BSs, since the BSs are the most expensive components (see Table 6.3), and the MIND-GO algorithm very effectively minimizes the fiber backhaul costs (see Figure 6.5 in Section 6.2). Moreover, it appears that the number of A-BSs remains approximately the same for all values of K (Figure 6.15), especially . This makes that the fiber costs is nearly a constant for different coverage parameter values in the same scenario. It must be noted that each W-BS should be within LoS of an A-BS. The entire map (all CSs) will be covered by A-BSs at some point. This is probably the reason for the constant number of A-BSs when $K \geq 3$. Furthermore, it must be true that not much extra bandwidth is assigned for backhaul links at A-BSs when K increases. This leads to the conclusion that the deployment of extra W-BSs does not result in significantly larger backhaul links. This agrees with the numbers in Figure 6.16, where we observe that the reserved backhaul capacity equals approximately $\frac{1}{K}$th of that reserved for access links. It is therefore expected that the increase of A-BSs would only further continue if either the used spectrum is diminished or the total demand of the TPs is enlarged, and not when K is set higher.

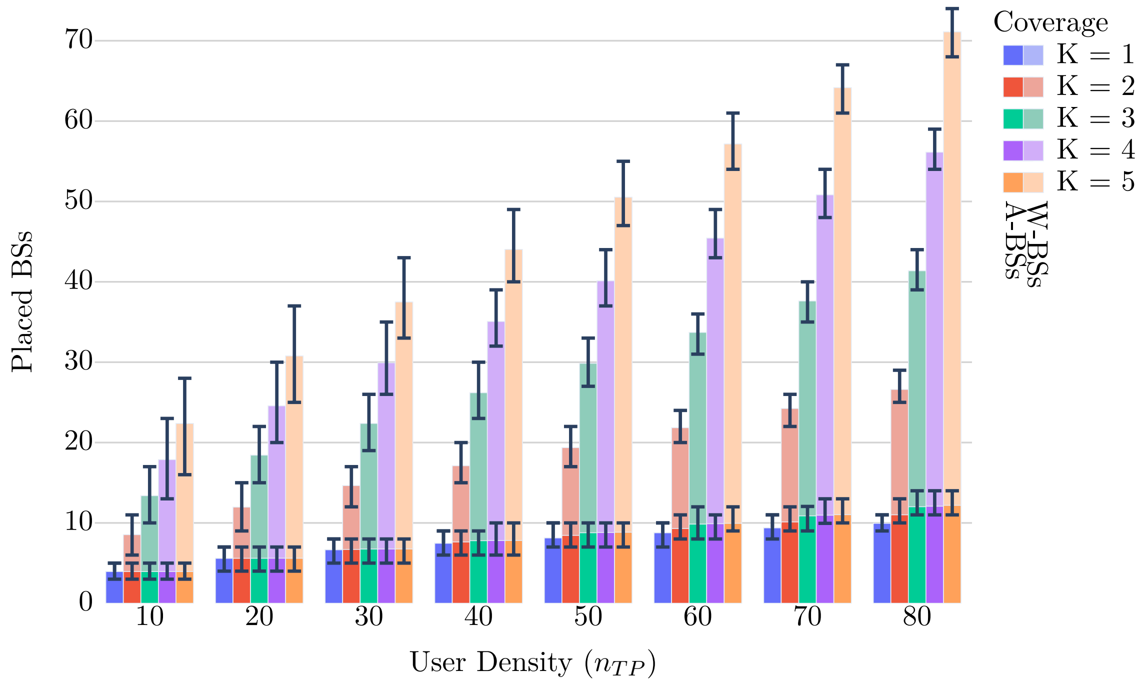

Figure 6.15: Placed BSs for increasing K and increasing n_{TP}

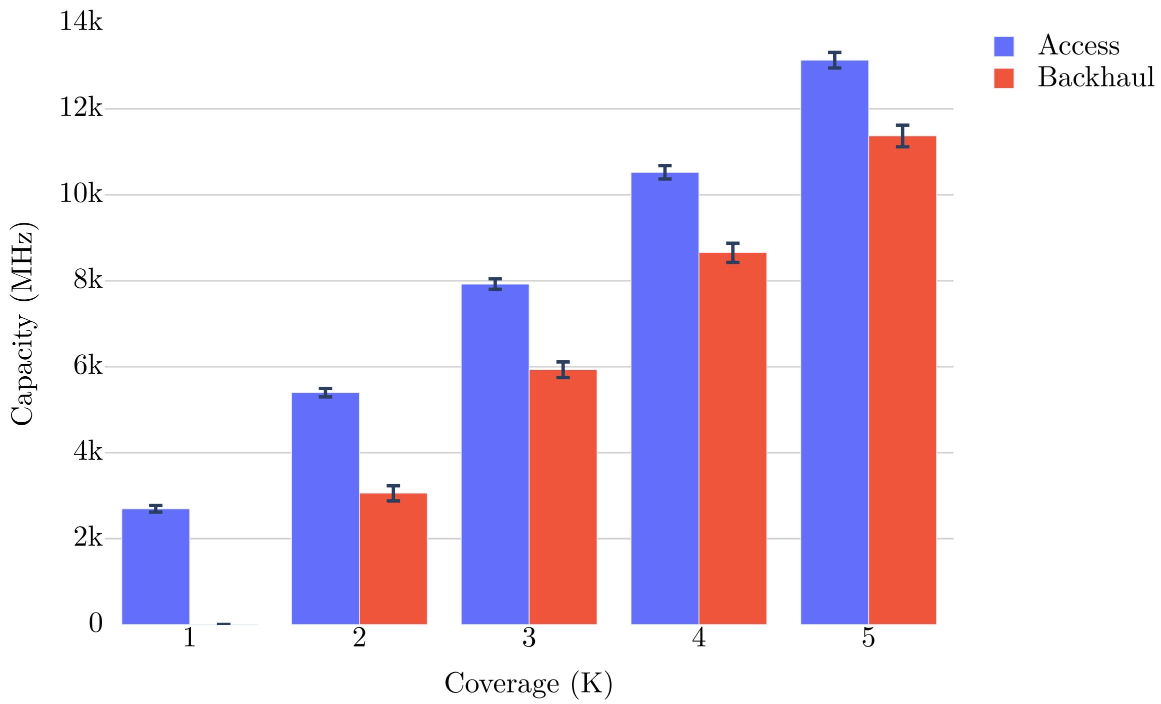

Figure 6.16: Allocated bandwidth capacity for access and backhaul links for $n_{TP} = 40$

Implications of MCo on system performance

Secondly, the AD&LL score is observed in Figure 6.18 to decrease with an increasing value of K. Before we analyze this, two remarks must be made. First, it should be noted that the system score for $K = 1$ is only determined by the average link length, and not by angular diversity. Secondly, the AD&LL metric is not designed to compare the access reliability for different values of K, but to compare different deployments with the same value of K. A deployment with a lower score but a higher K might therefore be a better deployment, depending on the system needs. For example, when the network should also support very precise localization, a higher value of K might be preferred. Additionally, which K is appropriate is very likely dependent on the number and size of the potential blockages.

Nevertheless, it is very interesting to analyze the decreasing score, since it appears from Figure 6.19 to be caused by a strongly declining AD score. We remind the reader that the AD score is normalized by $\frac{2\pi}{K}$, such that a score of 1 is theoretically possible for all values of K. The decreasing AD score might very well be influenced by the number of static blockages (like buildings) that play a larger role when K increases. This corresponds to our earlier observation in Section 6.4, that AD and LL scores converge when the open space chance increases. When TPs are on a road section with buildings on both sides, the minimal angle is also limited by the predetermined locations of CSs. This is illustrated in Figure 6.17.

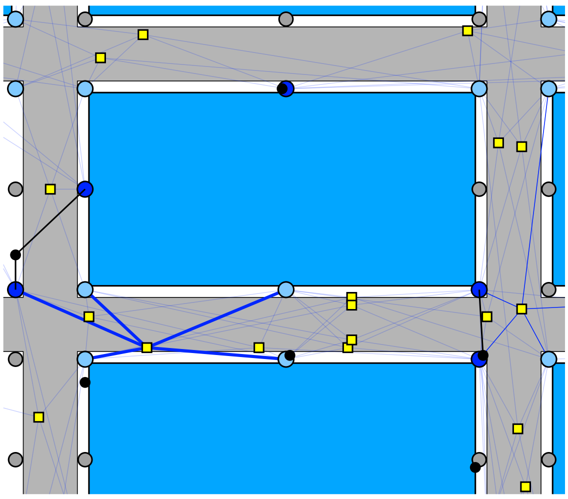

Figure 6.17: AD score is limited by buildings and CS locations, $K = 5$

Another interesting finding is that a higher number of TPs results in a higher score (Figure 6.18). This is presumably caused by the increased number of required BSs, which leaves more options for optimizing the AD&LL score.

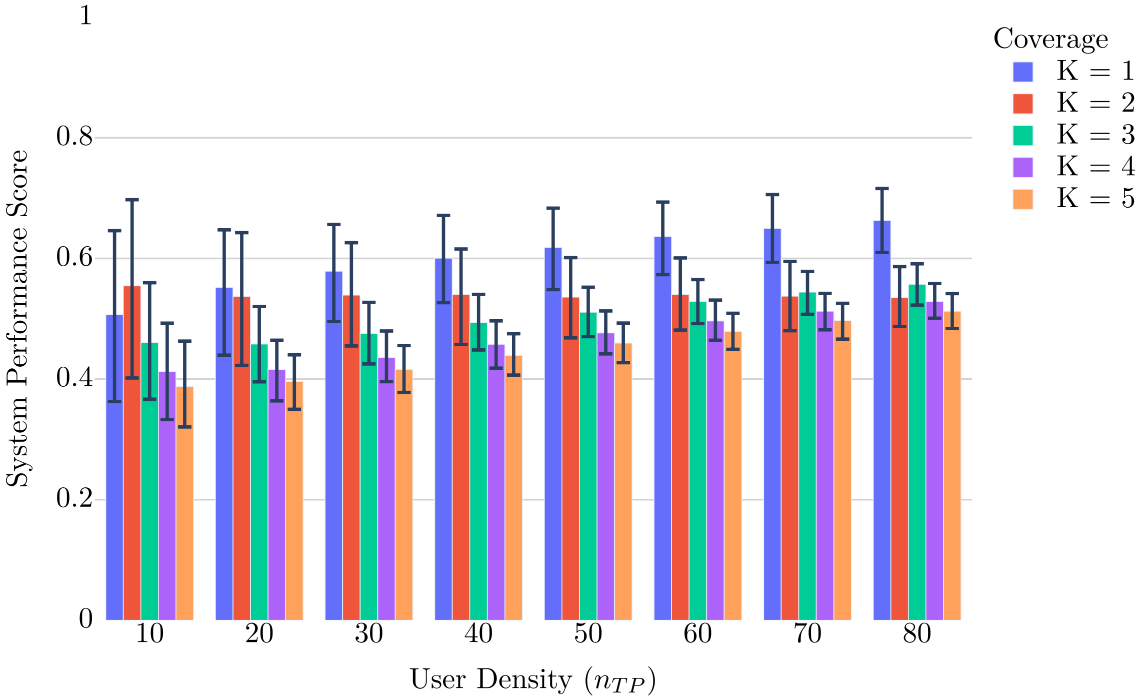

Figure 6.18: AD&LL scores for increasing K

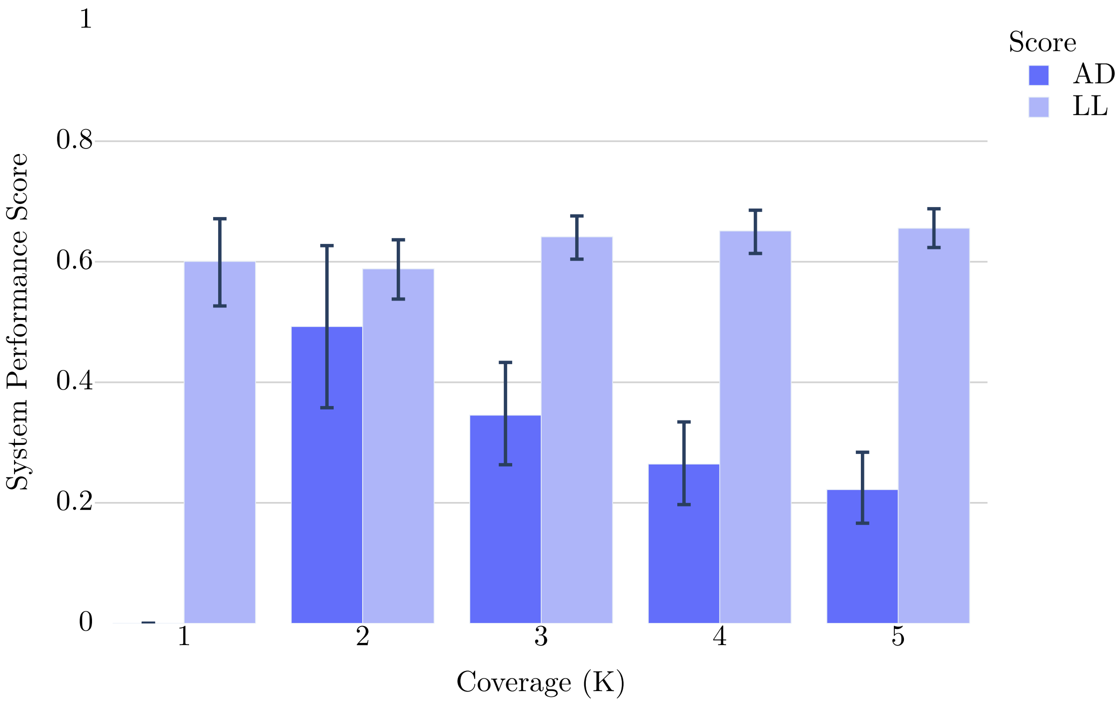

Figure 6.19: Split AD&LL scores for $n_{TP} = 40$ with increasing K

6.5.2 Implications of MCo on needed bandwidth for self-backhauling

In this subsection, an answer is sought to our second sub-question:

Q2. *What are the implications of using integrated access and backhauling on mmWave BS placement?*

For this, 1000 default maps are generated, on which measurements are done for user densities from 10 to 80 in steps of 10. Three algorithms are used to deploy BSs, namely the MIND-G algorithm (as described in Section 6.4), the MIND-GO algorithm, and a variant of the MIND-GO algorithm that only deploys A-BSs referred to as the "fully wired" algorithm. We report the average values of these runs for deployment costs, AD&LL scores, and assigned bandwidths, along with a 95% confidence interval. The results are presented to show the influence of self-backhauling (IAB) on (1) the system costs and (2) the system performance, respectively.

The fully wired algorithm executes the first two phases of our algorithm, using the same value for Γ^A in order to minimize the costs. One modification is made to the first phase. The MIND-GO algorithm namely divides the bandwidth of each A-BS equally over all covered TPs in the first phase. The fully wired algorithm, however, reserves bandwidth for full demand links at newly deployed A-BSs j considering the actual link lengths (L_{ij}) to its closest TPs i. Then, in the second phase, the links are reorganized such that obsolete A-BSs are removed, and then each TP is assigned the shortest possible K links.

Implications of IAB on deployment costs

The use of self-backhauling or IAB is aimed at reducing the network deployment cost. When this results in the need for much extra BSs to be deployed, the cost benefit might be outweighed. In Figure 6.20, it is observed that indeed more BSs are needed when IAB is used. However, because the deployment of A-BSs and accompanying fiber backhauls is much more costly, the deployment costs of the fully wired approach even exceeds the pure greedy MIND-G algorithm, as shown in Figure 6.21.

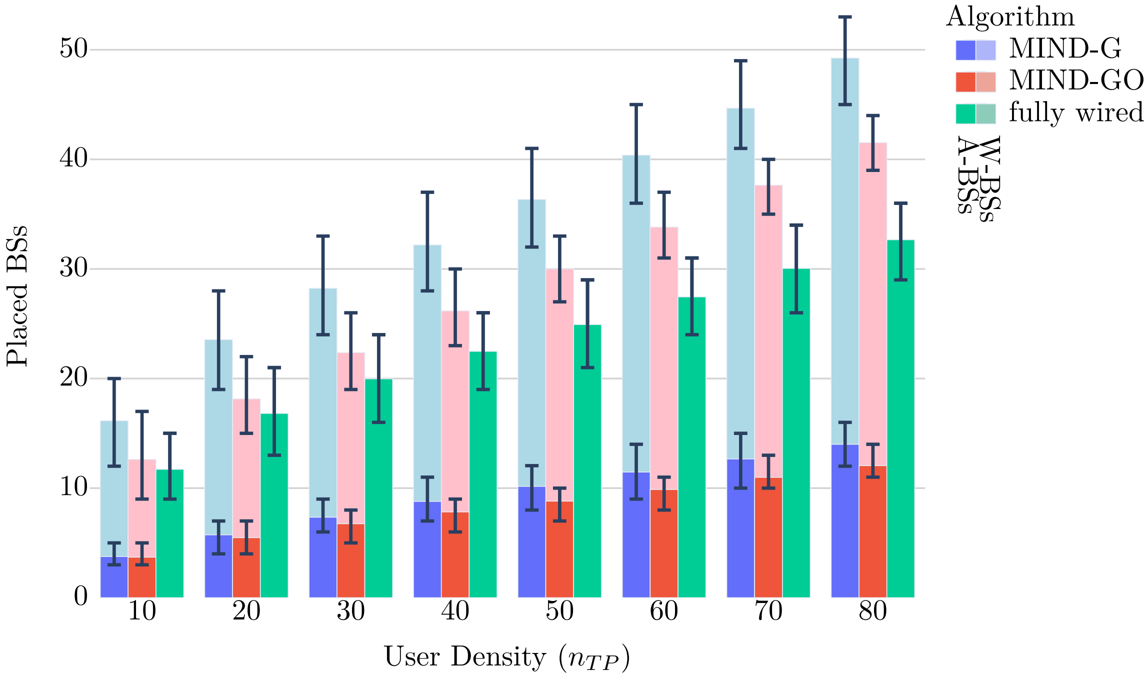

Figure 6.20: Placed BSs for

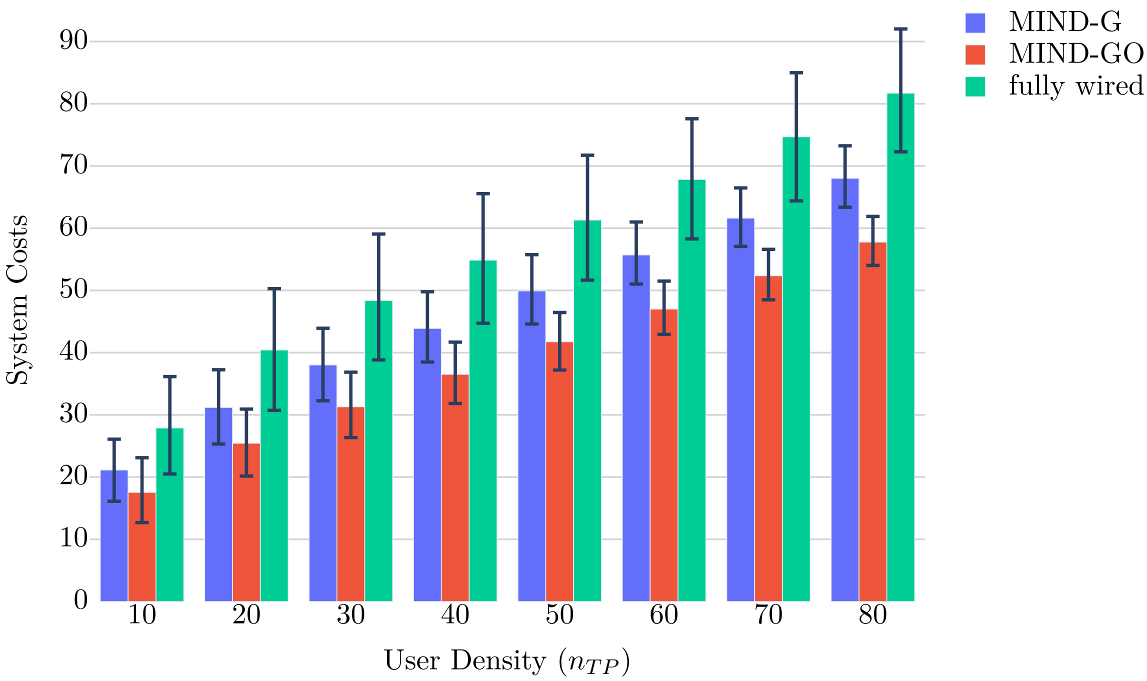

Figure 6.21: Split AD&LL scores for $n_{TP} = 40$ with increasing K

Implications of IAB on system performance

Figure 6.22 shows that the system performance of the fully wired algorithm is between 69% and 77% of the MIND-GO algorithm. This is only achieved by the second phase of the fully wired algorithm, which optimizes the link length (LL). This optimization is observed to even outperform the pure greedy MIND-G algorithm for the user densities $n_{TP} \leq 60$. The effect of optimizing a fully wired deployment for angular diversity (AD) as well has not been studied in this research. It is expected that this will increase the system costs, since this will require longer fiber links.

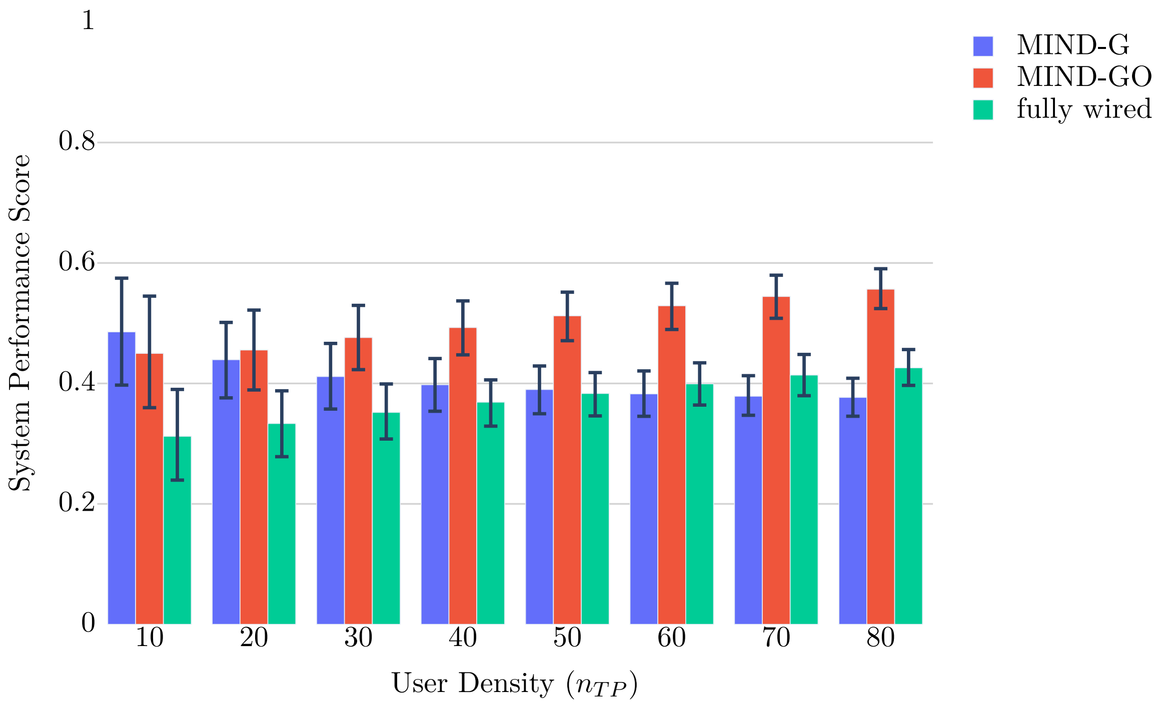

Figure 6.22: AD&LL scores for increasing K

6.6 Conclusions

In this chapter, we investigated various aspects of the MIND-GO algorithm. In Section 6.2, we determined the impact of scoring parameter weight values and chose values that result in the lowest deployment costs. We chose $\{0.5, 0.5, 0.0\}$ as weights for (1) fiber link length, (2) covered TPs, and (3) AD&LL score for A-BS deployment (Γ^A). For W-BS deployment, we set the scoring weight parameter (Γ^W) to $\{0.5, 0.5\}$, dividing the weight equally among (1) covered TPs and (2) AD&LL score. The main takeaway of this section is that some weight must be put on the fiber link length, since otherwise the deployment costs will significantly increase.

We found that the complexity of the two final optimization phases (phases 4 and 5) might be very well in $O(n_{TP}^3)$ instead of the expected $O(n_{TP}^2)$ in Chapter 5. This is polynomial, so it is still a good performance for a cell planning algorithm [10]. We argue that these phases should not be omitted, but suggest investigation of omitting only phase 4 if only deployment costs are important. The reason for not omitting the optimization phases is mainly that they improve the deployed network significantly compared to the results of only the first three phases. This is also further investigated as described below, using a pure greedy algorithm.

We compared the MIND-GO algorithm to two other algorithms in Section 6.4, namely a pure greedy algorithm and a brute force optimum algorithm. The results showed that the system costs are optimal in about 90% of the cases, and is only slightly improved in other cases (on average 4.2%). It is also apparent that the MIND-GO algorithm is superior to the pure greedy algorithm without optimization phases. Furthermore, it was observed that the angular diversity (AD) score improved when the open space chance was increased (Figure 6.14). This was not observed in the AD&LL scores (Figure 6.13), since the link length (LL) score could compensate for it.

We formulated answers to our research sub-questions in Section 6.5. We concluded that the use of multi-connectivity (MCo) with more links (higher K) leads to a higher number of placed W-BSs, but the use of self-backhauling (IAB) did not require significantly more A-BSs when K was increased. We also found that the AD score decreases significantly when K increases, whereas the LL score only slightly increases (less than 10%). Furthermore, the costs of using IAB evidently outweighs the costs of a fully wired deployment. This is tested for $K = 3$, and is expected to hold at least for higher values of K, but probably for $K = 2$ too.

From these results, we conclude that our algorithm is suitable for running simulations to gather design insights. Our current design insights are as follows.

- An environment with less static blockages enables a higher AD&LL score, which is mainly attributed to improvement of the AD partial score.
- In an environment with many static blockages, the AD partial score is dramatically decreased when K increases.
- Using IAB with MCo provides flexibility to increase the AD&LL score without increasing the system costs.
- Using IAB with MCo does not significantly increase the number of required A-BSs compared to the $K = 1$ scenario.

Finally, we conclude that using a combination of MCo and IAB is a cost-effective approach for mmWave CP.

Chapter 7

Conclusions and recommendations

Future smart city applications, e.g. intelligent transportation systems (ITS) will need mobile networks delivering throughputs in the order of around 1 Gbps with the additional requirement of ultra-reliable low latency communication (URLLC). This can be accomplished by mmWave communication. However, due to the high proneness to blockages, multi-connectivity (MCo) is needed to obtain appropriate access reliability. Using MCo however, increases the costs of the network deployment. To reduce these, integrated access and backhauling (IAB) can be used to decrease the amount of fiber connections necessary.

Furthermore, to improve the access reliability in mmWave networks with multi-connectivity, user equipments (UEs) should be served from multiple directions, to increase the chance for a reliable secondary link in case a blockage occurs. This so-called angular diversity (AD) is however limited by a combination of static blockages, such as buildings, and the the candidate site locations. The other factor determining the network's reliability, link length (LL), can be reduced to decrease the chance of dynamic blockages obstructing a link. Improving AD and reducing LL is possible when the number of base stations (BSs) is increased, which increases deployment cost. The combination of AD and LL is represented as the AD&LL score. This results in a trade-off between AD&LL and deployment cost. Understanding this trade-off is necessary when mmWave networks in an urban environment are being planned.

This trade-off was therefore investigated by designing a computationally efficient algorithm for mmWave network deployment with MCo and IAB. This algorithm was implemented in a simulator to find answers on our research questions. We first answer the main question on the placement of BSs, after which we answer the sub questions to investigate the influence of MCo and IAB in the simulated network deployment.

7.1 Answering the main research question

Our main question is stated as:

> *Given a set of candidate sites, at which of those locations should mmWave base stations be placed to provide the required coverage with appropriate quality of service for users in a city road environment while keeping costs as low as possible?*

To be able to build and investigate the algorithm that answers this question, we first defined a system that allowed us to model a city environment with test points (TPs), fiber access points (FAPs), and candidate sites (CSs). This is presented in Chapter 3. The CSs are a predetermined set of locations where a base station could be placed. An empirical channel model was used to model mmWave communication with beamforming. Interference between concurrent transmissions was not taken into account, as highly directional mmWave communication is considered noise-limited. The costs of BSs are mostly dependent on whether they are anchored and on the distance to the nearest FAP.

We formulated an optimization problem (Chapter 4), in which two functions define our objectives and all involved decision variables are defined. Our primary objective function is to minimize the deployment costs, including the costs of connecting BSs with FAPs. The secondary objective function is to minimize the blockage sensitivity, using the AD&LL score as a metric. The blockage sensitivity value is only decreased if this does not increase the deployment costs.

Then, we translated this optimization problem into an algorithm using a heuristic approach, as presented in Chapter 5. The algorithm prioritizes cost minimization. This is called the MIND-GO algorithm, which stands for MCo with IAB mmWave Network Deployment using Greedy construction with Optimization. We tested four aspects of this algorithm (Chapter 6). The conclusions from the first three aspects are summarized below. The final aspect is then discussed in Section 7.2.

Scoring CSs for BS deployment
Firstly, we tested how different weight parameter settings influence the deployment of both anchored and wireless BSs. These parameters are used to score CSs to determine where a BS should be placed. For the scoring weight parameter for A-BS deployment, we found that putting any weight on the scoring factor for the AD&LL score did not significantly improve the final system score. The reason for this is most probably that the optimization phases in our algorithm do influence the system score much more.

The lowest system cost was achieved by putting equal weights on the fiber cable length for A-BSs and on the number of TPs covered by a BS. However, the exact assignment of weights did not influence the system costs significantly, especially for large numbers of TPs.

Different values for the scoring weights parameter for W-BS deployment did not result in significant changes. Only when more than 60% of the weight was put on AD&LL score improvement, the costs increased slightly. Therefore, we chose to divide the weight equally between AD&LL score and the number of TPs covered by a newly placed W-BS.

Impact of each algorithm phase

Secondly, we investigated the influence of each of the five phase of the algorithm on the objective metrics. Two of these phases deploy BSs in a greedy way, and three phases optimize the deployment for our objective functions. We found that two of the optimization phases were by far the most time consuming, but that they also significantly reduced the deployment costs while keeping the AD&LL score high.

Algorithm comparisons

Thirdly, the performance of the MIND-GO algorithm in terms of the objectives was compared to two other algorithms, namely a brute force optimum and the pure greedy MIND-G algorithm (Section 6.4). For this, we used a very small setup to keep the running times for the brute force approach within reasonable limits. The MIND-G algorithm is used as a baseline algorithm, and executes only the two greedy phases from the MIND-GO algorithm, omitting the optimization phases. The brute force algorithm minimized the costs within the performance constraints.

From this comparison, it appeared that our algorithm performed on average within 90% of the optimal score, and that the deployment costs are significantly reduced compared to a purely greedy algorithm at the cost of additional computation time. Moreover, the confidence interval was much larger for the pure greedy algorithm.

Based on these findings, we conclude that our MIND-GO algorithm is a very suitable solution to run extensive simulations, on which design insights can be based.

7.2 Answering the research sub-questions

We ran multiple simulations to answer our two sub-questions, regarding the impact of multi-connectivity and integrated access and backhauling on optimal network deployment.

Multi-connectivity
First, our design insights on MCo are discussed, based on the sub-question:

Q1. *What are the implications of the requirement for multi-connectivity on mmWave BS placement?*

The answer to Q1 is determined by comparing the scores and costs of deployments with different values of K (for K-connectivity) resulting from our network planning algorithm. These simulations resulted in two design insights. Firstly, it appears that the required number of *anchored* base stations only increases when K is set to a value between 1 and 3. This might be explained by all candidate sites being covered by an A-BS when $K \geq 3$. It is expected that no more A-BSs are needed until backhaul links will become longer, resulting in the need for more reserved bandwidth per TP at A-BSs. Furthermore, the number of W-BSs to be deployed is found to increase linearly both with K and with the number of TPs. The required bandwidth per TP remains the same, so the total demand increases with the number of TPs. This means that costs per user of the deployment of an MCo network decreases when there are more users in the area.

Secondly, the value of the AD&LL score is observed to decrease with K. We suggest that this could be very well due to the restricted locations of CSs, combined with the presence of large static blockages (buildings) on both sides of many roads. It would therefore be interesting to further investigate the impact of a larger open space chance on the AD&LL score.

Integrated access and backhauling
In this section, our design insights on IAB are discussed, based on the sub-question:

Q2. *What are the implications of using integrated access and backhauling on mmWave BS placement?*

The answer to Q2 is determined by comparing the costs of deployments using IAB with deployments in which all BSs have a wired backhaul (A-BSs). It appears clearly that using IAB reduces the costs significantly, even though the number of deployed BSs is larger when IAB is used. The main reason for this is that an A-BS costs twice as much as wirelessly backhauled BSs (W-BSs), and the costs of the fiber backhaul deployment also increases. Additionally, the placement of W-BSs provides more flexibility to optimize the AD&LL score, since this does not influence the fiber deployment costs. Therefore, we conclude that using IAB is a very suitable way to reduce the deployment costs and optimize MCo against dynamic blockages.

7.3 Suggestions for future research

Based on experiences and results from our study, we provide here some directions for future research.

- **AD&LL influence on access reliability in realistic city traffic**
  Investigate the correlation of angular diversity and blockage probability in our scenario, with realistic moving obstacles. These might be moving buses and lorries on slow lanes as dynamic blockages, similar to how it is modeled by Tassi et al. in [2].

- **Best scoring weights for AD&LL metric for urban scenarios**
  It is then also interesting to study which weight ratio should be used between angular diversity and link length in the ad&LL score (parameter χ), since in our work χ is only based on the results of the original work by Devoti et al. [8]. In that study however, static blockages like buildings were not considered.

- **Reduce reserved bandwidth for secondary links**
  Explore the impact of reserving only a part of the needed capacity at W-BSs. This would only be interesting in a system where the bandwidth of W-BSs is more scarce, so probably other factors like the user demand must be adapted as well.

- **Correlation between open space chance and AD score**
  It would also be very interesting to know why the angular diversity is lower when K is larger. We posed the hypothesis that this is influenced by the low open space chance, but more simulations need to be run to validate this.

- **Multi-hop primary links**
  The system could be adapted such that primary links are not forced to connect directly to an A-BS, but that a W-BS might serve as additional hop. This way, more available capacity of the system could be used, while reducing the system costs even further by replacing a number of A-BSs by W-BSs.

- **MCo in the wireless backhaul network**
  The robustness of a deployed network could also be investigated. Since system components might fail anytime, for example due to an accident, it would be horrible if this would result in another accident because the V2X infrastructure would fail to provide sufficient quality of service. For this, the effect of multi-connectivity in the backhaul network on the network robustness against failures could be investigated.

References

[1] I. Mavromatis, A. Tassi, R. J. Piechocki, and A. Nix, “Efficient millimeter-wave infrastructure placement for city-scale ITS,” *arXiv*, 2019.

[2] A. Tassi, M. Egan, R. J. Piechocki, and A. Nix, “Modeling and Design of Millimeter-Wave Networks for Highway Vehicular Communication,” *arXiv:1706.00298 [cs, math]*, vol. 66, no. 12, pp. 10 676–10 691, 8 2017. [Online]. Available: http://arxiv.org/abs/1706.00298

[3] The 5G Infrastructure Public Private Partnership, “5G Automotive Vision,” *5G-PPP Initiative*, pp. 1–67, 2015. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-White-Paper-on-Automotive-Vertical-Sectors.pdf

[4] M. S. Elbamby, C. Perfecto, C.-F. F. Liu, J. Park, S. Samarakoon, X. Chen, M. Bennis, C. Perfecto, C.-F. F. Liu, J. Park, S. Samarakoon, X. Chen, and M. Bennis, “Wireless Edge Computing With Latency and Reliability Guarantees,” *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1717–1737, 8 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8734753

[5] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, R. W. Heath, N. González-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, “Millimeter-Wave Vehicular Communication to Support Massive Automotive Sensing,” *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 12 2016.

[6] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, “Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges,” *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62–68, 1 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7823339/

[7] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, “Tractable Model for Rate in Self-Backhauled Millimeter Wave Cellular Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2196–2211, 10 2015. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7110547

[8] F. Devoti and I. Filippini, “Planning mm-Wave Access Networks Under Obstacle Blockages: A Reliability-Aware Approach,” *IEEE/ACM Transactions on Networking*, pp. 1–12, 2020.

[9] A. L. Rezaabad, H. Beyranvand, J. A. Salehi, and M. Maier, “Ultra-Dense 5G Small Cell Deployment for Fiber and Wireless Backhaul-Aware Infrastructures,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 231–12 243, 2018.

[10] N. Palizban, S. Szyszkowicz, and H. Yanikomeroglu, “Automation of Millimeter Wave Network Planning for Outdoor Coverage in Dense Urban Areas Using Wall-Mounted Base Stations,” *IEEE Wireless Communications Letters*, vol. 6, no. 2, pp. 206–209, 2017.

[11] A. Asadi, S. Muller, G. H. Sim, A. Klein, M. Hollick, S. Müller, G. H. Sim, A. Klein, and M. Hollick, “FML: Fast Machine Learning for 5G mmWave Vehicular Communications,” in *Proceedings - IEEE INFOCOM*, vol. 2018-April, 4 2018, pp. 1961–1969.

[12] C. Campolo, R. Fontes, A. Molinaro, C. Rothenberg, and A. Iera, “Slicing on the Road: Enabling the Automotive Vertical through 5G Network Softwarization,” *Sensors*, vol. 18, no. 12, p. 4435, 12 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/12/4435

[13] L. Gavrilovska, V. Rakovic, and V. Atanasovski, “Visions Towards 5G: Technical Requirements and Potential Enablers,” *Wireless Personal Communications*, vol. 87, no. 3, pp. 731–757, 2016.

[14] I. Siaud, A.-m. U.-m. Orange, K. Safjan, P. Rugeland, M. Tercero, Y. Li, J. Luo, C. Fiandrino, J. Widmer, M. Filippou, H. Miao, A. Vijay, I. Siaud, A.-M. Ulmer-Moll, R. Li, M. Shariat, J. Lorca, and M. T. Aparicio, “Architectural enablers and concepts for mm-wave RAN integration,” pp. 1–26, 3 2017.

[15] M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, “Multi-connectivity in 5G mmWave cellular networks,” in *2016 Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*. Vilanova i la Geltru, Spain: IEEE, 6 2016, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/7528494/

[16] D. Kumar, J. Saloranta, J. Kaleva, G. Destino, and A. Tölli, “Reliable Positioning and mmWave Communication via Multi-Point Connectivity,” *Sensors*, vol. 18, no. 11, p. 4001, 11 2018. [Online]. Available: http://www.mdpi.com/1424-8220/18/11/4001

[17] G. Bielsa, J. Palacios, A. Loch, D. Steinmetzer, P. Casari, and J. Widmer, "Indoor Localization Using Commercial Off-The-Shelf 60 GHz Access Points," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. Honolulu, HI: IEEE, 4 2018, pp. 2384–2392. [Online]. Available: https://ieeexplore.ieee.org/document/8486232/

[18] G. R. Maccartney, T. S. Rappaport, and A. Ghosh, "Base Station Diversity Propagation Measurements at 73 GHz Millimeter-Wave for 5G Coordinated Multipoint (CoMP) Analysis," in *2017 IEEE Globecom Workshops, GC Wkshps 2017 - Proceedings*, vol. 2018-Janua, 12 2017, pp. 1–7.

[19] Y. Wang, A. Klautau, M. M. Ribero, A. C. K. Soong, and R. W. Heath, "MmWave Vehicular Beam Selection With Situational Awareness Using Machine Learning," *IEEE Access*, vol. 7, pp. 87 479–87 493, 2019.

[20] M. Giordani, M. Mezzavilla, and M. Zorzi, "Initial Access in 5G mmWave Cellular Networks," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 40–47, 11 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7744807/

[21] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Edge computing meets millimeter-wave enabled VR: Paving the way to cutting the cord," in *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 2018-April. Barcelona: IEEE, 4 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8377419/

[22] S. Carpenter, D. Nopchinda, M. Abbasi, Z. S. He, M. Bao, T. Eriksson, and H. Zirath, "A D-Band 48-Gbit/s 64-QAM/QPSK Direct-Conversion I/Q Transceiver Chipset," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 4, pp. 1285–1296, 4 2016.

[23] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, "IEEE 802.11ad: directional 60 GHz communication for multi-Gigabit-per-second Wi-Fi [Invited Paper]," *IEEE Communications Magazine*, vol. 52, no. 12, pp. 132–141, 12 2014.

[24] T. Bai, A. Alkhateeb, and R. W. Heath, "Coverage and capacity of millimeter-wave cellular networks," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 70–77, 9 2014.

[25] M. R. Akdeniz, Y. Liu, M. K. Samimi, S. Sun, S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter Wave Channel Modeling and Cellular Capacity Evaluation," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1164–1179, 6 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6834753

[26] A. I. Sulyman, A. M. T. Nassar, M. K. Samimi, G. R. MacCartney, T. S. Rappaport, and A. Alsanie, "Radio Propagation Path Loss Models for 5G Cellular Networks in the 28 GHz and 38 GHz Millimeter-Wave Bands," *IEEE Communications Magazine*, vol. 52, no. 9, pp. 78–86, 2014.

[27] H. Tabassum and E. Hossain, "Coverage and rate analysis for Co-existing RF/VLC downlink cellular networks," *arXiv*, vol. 14, no. 2, pp. 1100–1114, 2017.

[28] M. Gao, B. Ai, Y. Niu, Z. Zhong, Y. Liu, G. Ma, Z. Zhang, and D. Li, "Dynamic mmWave beam tracking for high speed railway communications," in *2018 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2018*. Barcelona: IEEE, 4 2018, pp. 278–283. [Online]. Available: https://ieeexplore.ieee.org/document/8368998/

[29] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, 4 2019. [Online]. Available: http://arxiv.org/abs/1804.01908

[30] G. P. P. P. A. W. Group, "View on 5G Architecture (Version 3.0)."

[31] A. Alkhateeb, I. Beltagy, and S. Alex, "Machine Learning for Reliable mmWave Systems: Blockage Prediction and Proactive Handoff," in *2018 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2018 - Proceedings*. Anaheim, CA, USA: IEEE, 11 2019, pp. 1055–1059. [Online]. Available: https://ieeexplore.ieee.org/document/8646438/

[32] Mmmagic, "Initial concepts on 5G architecture and integration: mmMAGIC Deliverable D3.1," mmMAGIC, Tech. Rep., 3 2016.

[33] O. Semiari, W. Saad, M. Bennis, and M. Debbah, "Integrated Millimeter Wave and Sub-6 GHz Wireless Networks: A Roadmap for Joint Mobile Broadband and Ultra-Reliable Low-Latency Communications," *IEEE Wireless Communications*, vol. 26, no. 2, pp. 109–115, 4 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8642794/

[34] 5G_PPP, "View on 5G Architecture," *Version 3.0, June 2019*, no. June, pp. 21–470, 2019. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2019/07/5G-PPP-5G-Architecture-White-Paper_v3.0_PublicConsultation.pdf

[35] M. Polese, M. Giordani, T. Zugno, A. Roy, S. Goyal, D. Castor, and M. Zorzi, "Integrated Access and Backhaul in 5G mmWave Networks: Potential and Challenges," *IEEE Communications Magazine*, vol. 58, no. 3, pp. 62–68, 3 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9040265/

[36] S. Saadat, D. Chen, and T. Jiang, "Multipath multihop mmWave backhaul in ultra-dense small-cell network," *Digital Communications and Networks*, vol. 4, no. 2, pp. 111–117, 4 2018. [Online]. Available: https://doi.org/10.1016/j.dcan.2017.08.002

[37] P. Muñoz, O. Adamuz-Hinojosa, P. Ameigeiras, J. Navarro-Ortiz, and J. J. Ramos-Muñoz, "Backhaul-Aware Dimensioning and Planning of Millimeter-Wave Small Cell Networks," *Electronics*, vol. 9, no. 1429, pp. 1–15, 2020.

[38] 5GPPP and G. A. W. P. P. P. A. W. Group, "View on 5G Architecture (Version 2.0)," *5Gppp Arrchitecture Working Group*, no. July, 1 2017.

[39] O. Teyeb, A. Muhammad, G. Mildh, E. Dahlman, F. Barac, and B. Makki, "Integrated access backhauled networks," *IEEE Vehicular Technology Conference*, vol. 2019-Septe, pp. 4–8, 2019.

[40] M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Max-Min Rates in Self-backhauled Millimeter Wave Cellular Networks," pp. 1–31, 2018. [Online]. Available: http://arxiv.org/abs/1805.01040

[41] A. Taufique, M. Jaber, A. Imran, Z. Dawy, and E. Yacoub, "Planning Wireless Cellular Networks of Future: Outlook, Challenges and Opportunities," *IEEE Access*, vol. 5, pp. 4821–4845, 5 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7883847/

[42] F. Luna, J. J. Durillo, A. J. Nebro, and E. Alba, "Evolutionary algorithms for solving the automatic cell planning problem: A survey," *Engineering Optimization*, vol. 42, no. 7, pp. 671–690, 2010.

[43] G. E. Athanasiadou, P. Fytampanis, D. A. Zarbouti, G. V. Tsoulos, P. K. Gkonis, and D. I. Kaklamani, "Radio network planning towards 5g mmwave standalone small-cell architectures," *Electronics (Switzerland)*, vol. 9, no. 2, pp. 1–10, 2020.

[44] Q. Wu, L. Chen, X. Chen, and W. Wang, "Cell planning for millimeter wave cellular networks," in *2017 9th International Conference on Wireless Communications and Signal Processing, WCSP 2017 - Proceedings*, vol. 2017-Janua. Institute of Electrical and Electronics Engineers Inc., 12 2017, pp. 1–6.

[45] G. Brown, "Exploring the potential of mmWave for 5G mobile access," *Heavy Reading*, no. June, 2016.