Dynamic Resource Allocation for UAV-Aided Networks

Salim Janji

Faculty of Computing and Telecommunications
Poznan University of Technology

This dissertation is submitted for the degree of
Doctor of Philosophy

Institute of Radiocommunications September 2025

عبثاً تحاول يا أبي مُلْكاً ومَمْلَكَةً
فَسِرْ للجُلْجُلَهْ
واصْعَدْ معي
لِنُعيْدَ للروح المُشَرَّد أوَّلَهْ
ماذا تُريد , وأنت سَيِّدُ روحنا
يا سَيِّدَ الكينونة المتحوّلَهْ؟
يا سَيِّدَ الجمرهْ
يا سَيِّدَ الشُّعْلَهْ
ما أوسع الثورة
ما أضيقَ الرحلة
ما أكبَرَ الفكرة
ما أصغَر الدولة

محمود درويش

In vain you strive, my father, for a realm and a kingdom,
so walk to Golgotha,
and climb with me,
that we might return the vagrant spirit to its first beginning.
What is it you seek you who are the master of our spirit,
O lord of the ever-shifting being?
O lord of the ember,
O lord of the flame,
How vast the revolution,
how narrow the journey,
how great the idea,
how small the state.

Mahmoud Darwish

To my uncle, mother, and grandparents who nurtured me with love and guided me with wisdom.
And to my beloved companion, whose presence brought strength, joy, and quiet constancy through every storm.
This journey, and this work, are as much yours as they are mine.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Salim Janji
September 2025

Acknowledgements

I would like to acknowledge the role that each of my teachers played throughout my academic journey, during which I was fortunate to meet and be influenced by dedicated scholars who were wholeheartedly committed to teaching and inspiring students to learn, grow, and continue along the path of education.

I begin with Prof. Rafic Ayoubi, who was the first to push the boundaries of my curiosity and learning in science, motivating me to improve my skills and encouraging me to follow the path of research. I was also fortunate to be supported by Prof. Krzysztof Wesołowski, my master's thesis supervisor, who helped me gain admission to the doctoral school at PUT. I further benefited greatly from my interactions with Dr. Adrian Kliks, whose influence shaped my perspective on science and academia. Finally, I will always deeply appreciate the support and guidance of Prof. Hanna Bogucka, who consistently provided me with positive outlooks and mentoring, especially during my doctorate. I would also like to thank Ms. Josephine Bahini, who always offered me support and motivation whenever it was needed.

I would also like to thank my colleagues and faculty members, many of whom I had the honor and pleasure of working with on different projects. I have learned much from each and every one of them, and for that I will always remain grateful.

Abstract

Drone base stations (DBSs) have emerged as promising components in next-generation wireless networks due to their mobility, altitude flexibility, and rapid deployment capabilities. These characteristics make drone base stations (DBSs) suitable for dynamic and cost-effective coverage in diverse scenarios, including disaster recovery, temporary events, and underserved regions. However, the deployment of DBSs introduces unique design challenges related to three-dimensional (3D) placement, energy limitations, wireless backhaul provisioning, and interference management.

This thesis presents a comprehensive framework for the design and optimization of DBS-assisted wireless networks, focuing on aspects specific to the DBS deployment. First, the thesis introduces the foundational concepts, regulatory considerations, and classifications of unmanned aerial vehicles (UAVs) relevant to network integration. Then, a modular simulation environment is developed to evaluate proposed algorithms under varied urban topologies, user mobility models, energy harvesting mechanisms, and communication channel characteristics.

Five core contributions are made across fronthaul and backhaul dimensions. For fronthaul optimization, novel DBS placement strategies are introduced for both known and unknown ground nodes (GNs)' locations using signal-to-interference-plus-noise ratio (SINR)-based clustering and multi-agent reinforcement learning, respectively. A Monte Carlo tree search (MCTS) algorithm is then developed for adaptive transmission power control, improving link reliability and energy efficiency, and enabling cell deactivation. When considering backhaul aspcet and discarding fronthaul optimization, the thesis proposes a visibility-graph-based placement framework that incorporates solar energy harvesting and reconfigurable intelligent surfaces (RISs) to ensure line of sight (LOS) connectivity between cells levaraging directive channel technologies. Then, to jointly optimize both fronthaul and backhaul aspects, a joint placement strategy is formulated using hierarchical clustering (HC) to satisfy simultaneous fronthaul coverage and backhaul mesh connectivity. Finally, a genetic algorithm (GA) is introduced to solve the drone network problem (DNP), optimizing the wireless mesh topology, formulated as a graph, under flow and degree constraints. Thus, the work provides a comprehensive deployment framework that optimizes the fronthaul and backhaul metrics, and is capable of adapting to network changes.

The proposed solutions demonstrate significant gains in energy efficiency, scalability, and throughput. Together, they form intelligent AI-driven solutions for 3D DBS-assisted networks.

Streszczenie

Stacje bazowe na pokładzie dronów (ang. drone base stations (DBSs)) mogą być obiecującymi elementami sieci radiowych nowej generacji dzięki swojej mobilności, elastyczności w zakresie wysokości operacyjnej oraz możliwości szybkiego wdrożenia. Cechy te sprawiają, że DBSs są odpowiednie do dynamicznego i opłacalnego zapewniania zasięgu w różnorodnych scenariuszach, takich jak sytuacje kryzysowe, wydarzenia tymczasowe czy brak stałego pokrycia usługami telekomunikacyjnymi (w tzw. regionach wykluczonych cyfrowo). Jednak ich wdrażanie wiąże się z unikalnymi wyzwaniami projektowymi, dotyczącymi rozmieszczenia w przestrzeni trójwymiarowej, ograniczeń energetycznych, zapewnienia łączy dosyłowych oraz zarządzania interferencją.

Niniejsza praca przedstawia kompleksowe podejście do projektowania i optymalizacji sieci z wykorzystaniem DBSs, koncentrując się na aspektach specyficznych dla ich wdrożenia. W pierwszej kolejności omówiono podstawowe pojęcia oraz uwarunkowania regulacyjne bezzałogowych statków powietrznych (dronów) (ang. unmanned aerial vehicles (UAVs)) istotne dla ich integracji z infrastrukturą sieci telekomunikacyjnych z dostępem radiowym. Następnie opracowano środowisko symulacyjne umożliwiające ocenę proponowanych algorytmów w różnych topologiach miejskich, modelach mobilności użytkowników, mechanizmach pozyskiwania energii oraz charakterystykach kanałów komunikacyjnych.

W pracy przedstawiono pięć głównych osiągnięć naukowych autora, obejmujących aspekty łączy dostępowych i dosyłowych. W zakresie optymalizacji łączy dostępowych zaproponowano nowe strategie rozmieszczania DBSs dla przypadków znanych i nieznanych lokalizacji węzłów naziemnych (ang. ground nodes (GNs)), odpowiednio z wykorzystaniem grupowania opartego na wartościach stosunku mocy sygnału do interferencji (ang. signal-to-interference-plus-noise ratio (SINR)) oraz wieloagentowego uczenia ze wzmocnieniem. Następnie opracowano zastosowanie algorytmu przeszukiwania drzewa wariantów Monte Carlo (ang. Monte Carlo tree search (MCTS)) do adaptacyjnego sterowania mocą transmisji, poprawiający niezawodność łączy i efektywność energetyczną oraz umożliwiający dezaktywację komórek. Oprócz optymalizacji warstwy dostępowej, zaproponowano technikę rozmieszczania DBSs opartą na grafach widoczności, uwzględniającą pozyskiwanie energii słonecznej oraz zastosowanie inteligentnych matryc odbiciowych (ang. reconfigurable intelligent surfaces (RISs)) w celu zapewnienia łączności bezpośredniej (w linii widzenia) między komórkami z wykorzystaniem technologii kierunkowych. W celu jednoczesnej optymalizacji warstw dostępowej i dosyłowej sformułowano wspólną strategię rozmieszczania węzłów sieci opartą na algorytmie hierarchicznego grupowania (ang. hierarchical clustering (HC)), zapewniającą zarówno zasięg dostępowy, jak i siatkę połączeń dosyłowych. Na koniec

wprowadzono algorytm genetyczny (ang. genetic algorithm (GA)) do rozwiązania problemu optymalizacji topologii sieci dronowej (drone network problem (DNP)) zdefiniowanej w formie grafu, przy uwzględnieniu ograniczeń jego przepływu i stopnia.

Proponowane metody wykazują znaczące zyski pod względem efektywności energetycznej, skalowalności oraz odporności na przeciążenia. Łącznie stanowią inteligentne, oparte na artificial intelligence (AI) rozwiązania dla trójwymiarowych sieci wspieranych przez DBSs.

Contents

List of Figures

List of Tables

Symbols

Roman Symbols

G_r	Transmitter antenna gain
G_R	RIS antenna element gain
G_t	Transmitter antenna gain
M	Number of DBSs or DRSs
U	Number of GNs
B	Number of terrestrial base stations (BSs) and gateway nodes, both terrestrial and non-terrestrial
C_{D0}	Profile drag coefficient of DBS flight
P_{DBS}	Total aerodynamical power consumption of a DBS in W
P_{drag}	Power consumption required for drag flight of DBS in W
P_{level}	Power consumption required for level flight of DBS in W
(v_x, v_y)	Horizontal velocity components characetrizing DBS movement in $\mathrm{m\,s^{-1}}$
$d_{2\mathrm{d}}(j,n)$	The shortest distance between two points indexed by j and n in a two-dimensional space, measured along the xy-plane in m
$d(j,n)$	The shortest distance between two points indexed by j and n in a three-dimensional space, measured in a straight line in m
A_D	Drone's total rotor disk area in m^2
$\boldsymbol{M}(n)$	SINR-based decoupling mask vector for GN n
P_j^{EM}	Mixing proportion of cluster j in Gaussian mixture model
S_{th}	Dynamic SINR threshold for the SINR-based EM clustering approach (SINR-EM) algorithm in dB
$s(\boldsymbol{z})$	Softmax function applied to vector $\boldsymbol{z}$
f_c	Carrier frequency in Hz
$G(j)$	Total GN load serviced by DBS j
$W'(j,n)$	Binary variable mapping the max-signal-to-noise ratio (SNR) association between GNs and DBSs
$R_{\mathrm{A}}(j)$	Coverage disk of DBS j resulting from SNR threshold

$b(i,l)$	Binary variable indicating whether BS l is the forwarding hop of BS i
$f(i,l)$	Flow variable which represents the total downlink load that DBS i transmits to its forwarding hop BS l
h_0	Initial altitude value for all DBSs
$\mathrm{PL}_{\max}$	Maximum path loss value constraint that must be guaranteed between a GN and its serving DBS
N_{B}	Minimum required number of DBS backhaul neighbors
$a(k)$	Binary selection variable determining if DBS k from the set $\mathcal{A}$ is deployed
$R_{\mathrm{A}}(j)$	Coverage radius of a DBS for a given $L_{\max}$ constraint in m
$F(t)$	Total fairness score at time instance t
$F_j(t)$	Fairness score obtained for DBS j at time instance t
T_F	Moving time window size for calculating the fairness metric in s
$W(j,n)$	Binary variable mapping the max-SINR association between GNs and DBSs
$L_{\max}$	Maximum path loss (PL) threshold for the GN-DBS channel
$R_{\max}$	Maximum achievable coverage radius of a DBS, given the maximum allowable path loss and height constraints.
$R(j,n)$	Capacity of the channel between GN n and DBS j in $\mathrm{bit\,s^{-1}}$
$P_R(j,n)$	Power received by GN n from DBS j in W
t_s	Time step between subsequent fronthaul channel measurements updates in s
N_j	Number of GNs associated with DBS j and achieving a minimum SINR value of Γ_{th}
N	Total number of GNs achieving a minimum SINR value of Γ_{th}
P_{total}	Total transmission power summed across all DBSs in W
R_{total}	Total throughput achieved by all GNs in $\mathrm{bit\,s^{-1}}$
$P_T(j)$	Fronthaul transmission power of DBS j in W
$\boldsymbol{P}_T$	Vector containing fronthaul transmission power settings of DBSs in W
B_n	Fronthaul channel bandwith assigned to GN n in Hz
$\mathrm{cov}_n(t)$	Coverage score of GN n at time instance t
B_0	Fixed fronthaul channel bandwidth assigned to all GNs in Hz
h_a	Atmospheric turbulence-induced fading of free-space optics (FSO) channel
h_p	Atmospheric loss of FSO channel
h_g	geometric and misalignment losses (GML) loss of FSO channel
$d_{\max}$	FSO maximum link distance in m
P_{FSO}	FSO transmission power in W
A_0	Maximum fraction of power captured by FSO receiver
$R_{\mathrm{FSO}}(i,l)$	FSO link rate between BSs i and l in $\mathrm{bit\,s^{-1}}$

R Uniform fronthaul throughput required assumed for all GNs

R_n Fronthaul throughput required by a GN n

$\boldsymbol{m}_j = (x_j, y_j, h_j)$ Vector denoting the three-dimensional (3D) location of DBS j

$\boldsymbol{u}_n = (x_n, y_n)$ Vector denoting the 2D location of GN n

$H_H(x)$ Heaviside step function.

C_{BS} Expected capacity of a DBS given its bandwidth and system's spectral efficiency, in in $\text{bit}\,\text{s}^{-1}$

$U_{\max}$ Maximum number of GNs served by a DBS given its expected capacity and GN assigned bandwidth

$S_{\text{node}}^{(i)}$ Score of a node in the MCTS tree at iteration i

$S_{\text{win}}^{(i)}$ Score of winning child node in the MCTS tree at iteration i

$V^(i)_{\text{node}}$ Visits count of MCTS tree node at iteration i

N_{th} Required number of GNs to be served, based on total users U and link reliability threshold Γ_{th}

t_{update} Time step between subsequent MCTS updates in s

$N_{\text{node}}^{(i)}$ Total number of GNs achieving a minimum SINR value of Γ_{th} for a given MCTS tree node at iteration i

UCB_{child} Upper confidence bound score of a child node in TP-MCTS

K_e Exploration parameter of the UCB score in TP-MCTS

$L_{\text{mm}}^{[\text{dB}]}(i,l)$ Path loss (in dB) of millimeter-wave (mmWave) link between BSs i and l in $\text{bit}\,\text{s}^{-1}$

N_{TCP} Number of clusters in a Thomas cluster process

N_R RIS number of unit cell columns

M_R RIS number of unit cell rows

$F(\theta)$ Radiation pattern gain of RIS element in direction of θ

X_{RIS} The count of meta-surfaces within the RIS

d_x, d_y Dimensions of the RIS element

$L_{\text{RIS,B}}^{[\text{dB}]}(i,l)$ Path loss of link established through a RIS between BSs i and l in dB

A_R Amplitude of the reflection coefficient of RIS element

G_t Expected cumulative discounted reward starting from state $\boldsymbol{s}_t$ up to a terminating state

T_{epoch} Time duration of a Q-learning epoch

$Q^{\pi}(\boldsymbol{s}, \boldsymbol{a})$ Q function characterizing the expected cumulative discounted reward return if action $\boldsymbol{a}$ is taken given state $\boldsymbol{s}$

$V^{\pi}(\boldsymbol{s})$ State value function assuming policy π within the Q-learning framework

I_0 Average solar irradiation in $\text{W}\,\text{m}^{-2}$

T_{atm}	Atmospheric solar transmittance
c	Speed of light in $\mathrm{m\,s^{-1}}$
$G_b(\varphi_b, \theta_b)$	Beamforming gain given azimuth and elevation angles of the beam.
s	Log-normal shadowing sample added to path loss calculations.
G_t	Transmitter gain for V2X beamforming model.
G_o	Graph describing the set of geometric objects (obstacles) in an urban area
G_v	Graph containing all possible edges with clear line of sight (LOS) between any two nodes in G_o

Greek Symbols

Ω_W	Drone weight in N
ρ_a	Air density in $\mathrm{kg\,m^{-3}}$
ω	Elevation angle from GN to DBS in degree
$\pi(\boldsymbol{u};\theta)$	Likelihood of sample $\boldsymbol{u}$ under the GMM with parameters θ
$\hat{\Gamma}_{\text{th}}^{(l)}(n)$	GN-specific SINR threshold at iteration (l) in the SINR-EM algorithm
η	Average excessive loss of propagation group in dB
α_{mm}	Path loss exponent for mmWave channel
α_{RIS}	Path loss exponent for a RIS-based channel
ζ	Achieved energy efficiency in bit/J
$\Phi(t)$	The total link reliability defined as the ratio of number of GNs achieving Γ_{th} to total number of GNs
$\Phi_j(t)$	The link reliability per DBS j defined as the ratio of number of GNs achieving Γ_{th} to total number of GNs associated to DBS j
σ_0	GNs's receiver noise power in W
ω_{opt}	Optimal elevation angle relating a DBS's height to its maximum coverage radius in a given environment
$\Gamma(j,n)$	Received SINR on the channel between GN n and DBS j
Γ_{th}	Minimum SINR threshold for service in linear scale
$\Gamma_{\text{th}}^{\text{dB}}$	Minimum SINR threshold for service in dB
$\boldsymbol{\Gamma}(n)$	Vector containing SINR values measured by GN n from all DBSs in dB
ω_0	FSO beam waist radius in m
$\omega(d)$	FSO beam width in m
$\rho(d)$	FSO beam coherence length in m
κ	FSO channel weather-dependent attenuation coefficient in $\mathrm{m^{-1}}$
λ_1, λ_2	Parameters characterizing FSO GML due to angle variations
σ_F	Noise power at FSO receiver in W
η_{FSO}	Responsivity of FSO photo-detector

λ_o	FSO optical wavelength in m
ζ_{FSO}	Parameter characterizing bounds on FSO GML loss
$\varkappa$	Scaling constant for the Softmax function
λ_{TCP}	Poisson intensity of parents in a Thomas cluster process (TCP)
λ_{U}	Poisson intensity of number of childs in TCP
$\zeta_{\text{node}}^{(i)}$	Achieved energy efficiency of MCTS tree node at iteration i in bit/J
Φ	Desired link reliability defined as the ratio of number of GNs achieving Γ_{th} to total number of GNs
$\boldsymbol{\mu}$	two-dimensional (2D) xy coordinates of cluster center (mean)
α	Environment-dependent variable for path loss calculation
β	Environment-dependent variable for path loss calculation
λ	Discount factor of future rewards within the Q-learning approach
ε	Exploration probability of Q-learning approach
γ	Learning rate of the Q-learning approach
Σ	Covariance matrix of a multivariate distribution
σ^2	Cluster variance value across each dimension
α_{solar}	Solar altitude angle in degree
ϕ_{solar}	Solar azimuth angle in degree
δ_{cloud}	Uniform random variable characterizing solar attenuation caused by clouds
η_{PV}	photovoltaic panels (PV) panel efficiency factor
S_{PV}	PV panel surface area in m^2
γ_{SE}	System average spectral efficiency in $\text{bit}\,\text{s}^{-1}\,\text{Hz}^{-1}$
α_U	Ratio of built-up land area to the total land area
β_U	Mean number of buildings per unti area
γ_U	Scale parameter of Rayleigh distribution of building heights in m
φ_b	Beamwidth azimuth angle for the beam covering vehicular users
θ_b	Beamwidth elevation angle for the beam covering vehicular users
λ_v	Standard deviation of the log-normal distribution characterizing shadowing in vehicular beamforming.
σ_s	Standard deviation of the log-normal distribution characterizing shadowing in vehicular beamforming.
λ_{wave}	Wavelength of transmitted signal in m

Other Symbols

$\mathcal{K}_j$	Possible locations for DBS j
$\mathcal{A}$	Set of total DBSs available for deployment

$\mathcal{N}(\boldsymbol{u}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ Probability of observing a sample at $\boldsymbol{u}$ given the multivariate normal distribution with mean $\boldsymbol{\mu}_i$ and covariance matrix $\boldsymbol{\Sigma}_i$

$\mathcal{A}_{\mathrm{Q}}$ Action space of an agent within a given Markov decision process (MDP)

$\mathcal{S}_{\mathrm{Q}}$ State space of a given MDP

$\mathcal{B}$ Set comprising terrestrial BSs and gateway nodes, both terrestrial and non-terrestrial

$\mathcal{E}$ Set comprising the potential backhaul connections for the BSs in $\mathcal{V}$

$\mathcal{V}$ Comprehensive set comprising both $\mathcal{M}$ and $\mathcal{B}$

$\mathcal{M}$ Set of DBSs

$\mathcal{U}$ Set of GNs

$\mathcal{P}_A^{\mathrm{new}}$ Vehicular access probability according to the new model proposed in this thesis.

$\mathcal{P}_A^{\mathrm{old}}$ Access probability according to the old model presented in literature.

$\mathcal{P}_{\mathrm{out}}$ Outage probability for vehicles covered with a beam.

$\mathcal{P}_{\overline{\mathrm{res}}}^{\mathrm{new}}$ Probability of lacking sufficient bandwidth according to the novel model presented in this thesis.

$\mathcal{P}_{\overline{\mathrm{res}}}^{\mathrm{old}}$ Probability of lacking sufficient bandwidth according to the old model presented in literature.

Acronyms

2D two-dimensional
3D three-dimensional
3G third generation
4G fourth generation
5G fifth generation
6G sixth generation

A2A air-to-air
AI artificial intelligence
ANN artificial neural network
AWGN additive white Gaussian noise

B5G beyond fifth generation
bps bits per second
BRS best root selection
BS base station

CAPEX capital expenditure
CDF cumulative distribution function
CL complete linkage method for agglomerative hierarchical clustering

D2D device-to-device communications
DBS drone base station
DBSCAN density-based spatial clustering of applications with noise data clustering algorithm
DDPG deep deterministic policy gradient algorithm
DL deep learning
DNP drone network problem
DQL deep Q-learning
DRS drone relay station

EASA European Union Aviation Safety Agency

EE energy efficiency
EM expectation-maximization

FAA Federal Aviation Administration
FANET flying ad-hoc network
FDMA frequency-division multiple access
FSO free-space optics
FSPL free-space path loss

GA genetic algorithm
GIS geographic information system
GML geometric and misalignment losses
GMM Gaussian mixture model
GN ground node
GSC geometric set cover problem

HAP high altitude platform
HC hierarchical clustering
HPP Hamiltonian path problem

IOT internet of things
IP internet protocol
ITU International Telecommunication Union

k-means++ k-means clustering algorithm with k-means++ seeding

LAP low altitude platform
LOS line of sight

MBS terrestrial macro base station
MCTS Monte Carlo tree search
MDP Markov decision process
MILP mixed-integer linear program
MIMO multiple-input multiple-output
mmWave millimeter-wave

NLOS non-line of sight
NR New Radio

OFDMA orthogonal frequency-division multiple access

OMM obstacle mobility model
OOP object-oriented programming
OPEX operational expenditure
OSM OpenStreetMap

PCP Poisson cluster process
PDF probability density function
PL path loss
PMF probability mass function
PPP Poisson point process
PSO particle swarm optimization algorithm
PV photovoltaic panels

QOS quality of service
QoS quality of service

RAN radio access network
RF radio frequency
RIS reconfigurable intelligent surface
RL reinforcement learning
RWP random waypoint mobility model

SC small cell
SINR signal-to-interference-plus-noise ratio
SINR-EM SINR-based expectation-maximization (EM) clustering approach
SNR signal-to-noise ratio
SWIPT simultaneous wireless information and power transfer

TCP Thomas cluster process
TDMA time-division multiple access
TP-MCTS transmission power Monte Carlo tree search

UAV unmanned aerial vehicle
UCB upper confidence bound score
UE user equipment
UPGMC unweighted pair group method using centroids for agglomerative hierarchical clustering

VLOS visual line of sight

WMN wireless mesh networks

Notations

The following list gives an overview of some specific notation used throughout this manuscript. Boldface letters ($\boldsymbol{M}$, $\boldsymbol{m}$) are used for matrices or vectors, and regular letters for scalars (a). Sets are represented by calligraphic uppercase letters ($\mathcal{K}$).

$(a_1, \ldots, a_N)$	N-tuple
$\arg\max$	The domain element that maximizes the function.
$\arg\min$	The domain element that minimizes the function.
$\max\{\cdot\}$	maximum value within a set
$\min\{\cdot\}$	maximum value within a set
$\lceil \cdot \rceil$	Ceiling function
$\lfloor \cdot \rfloor$	Floor function
$\mathbb{B}$	Boolean binary domain $\{0, 1\}$
$\mathbb{E}_X[\cdot]$	Expectation over X
$\mathbf{A}^T$	Transpose of matrix $\mathbf{A}$
$\mathbf{V}^T$	Transpose of vector $\mathbf{V}$
$\odot$	element-wise multiplication of two vectors or matrices
$\|\| \cdot \|\|$	L2 norm of a vector
$\binom{k}{n}$	Binomial coefficient for selecting n out of k elements.

Preface

Since the 2010s, unmanned aerial vehicles (UAVs), also known as drones, have transitioned from primarily military tools to widespread civilian use, continuously evolving and adapting to commercial applications such as aerial photography, disaster response, delivery services, infrastructure inspections, and, more recently, collecting sensing data and providing wireless connectivity. Several factors have accelerated the widespread adoption of drones, including advances in battery power efficiency, the development of solar panels to extend mission durations, the introduction of flight automation and GPS navigation systems, and the design of lightweight, high-resolution sensors and cameras that enable obstacle avoidance and object tracking. Another key factor is the introduction of regulations by government bodies governing the civilian use of UAVs [1].

The mobility of drones is what makes them attractive for many applications. In the context of providing wireless connectivity, which is the topic of this work, drone mobility presents unique advantages, such as the ability to dynamically adjust their positions based on the distribution and traffic of ground nodes (GNs), thus establishing line of sight (LOS) connectivity and improving spectral efficiency. Furthermore, drone base stations (DBSs) operating at high altitudes can more easily establish directive LOS links with other aerial and terrestrial cells due to the increased LOS probability. Another advantage is that a DBS network enables quicker replacement and maintenance, as a drone can quickly return to a maintenance center for repairs or battery replacement, minimizing downtime.

DBSs can also be deployed quickly and provide coverage over various types of terrain, making them an appropriate solution for providing connectivity to search and rescue teams and disaster-struck areas. Furthermore, in scenarios where a permanent infrastructure is unnecessary, DBSs offer a cost-effective alternative to deploying fixed terrestrial base stations (BSs). The expenses incurred by conventional cellular operators include the costs of acquiring and constructing a site, referred to as capital expenditure (CAPEX), and recurring costs for maintenance, personnel, and electricity, known as operational expenditure (OPEX). DBSs deployment can significantly reduce both CAPEX and OPEX costs [2].

Cellular networks have undergone significant transformations in performance and architecture since the 1980s, with each new generation driven by key technologies that introduce new advantages, challenges, and design principles. Advancements in radio frequency (RF) electronics enabled frequency-division multiple access (FDMA), while improvements in clock precision and synchronization led to time-division multiple access (TDMA). The development of coding, modulation, and digital signal processing techniques facilitated the transition to third generation (3G) networks, which also introduced multiple-input multiple-output (MIMO) techniques. These methods were further extended in fourth generation (4G) networks, incorporating massive MIMO, orthogonal frequency-division multiple access (OFDMA), and internet protocol (IP)-based networking to enhance spectral efficiency and data throughput. Currently, fifth generation (5G) networks introduce beamforming and millimeter-wave (mmWave) transmission, improving network capacity and reducing latency. Looking ahead, beyond fifth generations (B5Gs) and sixth generations (6Gs) networks will be driven by artificial intelligence (AI)-based network optimization and self-optimization, energy-efficient communication leveraging ambient energy, and the integration of satellite and UAVs into terrestrial networks [3]. The incorporation of these new technologies and their impact on the architecture of future networks, especially considering the mobility of DBSs, remain open questions. This thesis examines these challenges, focusing on the role of DBSs in next-generation networks and their potential to reshape wireless communication paradigms.

This thesis addresses the key challenges associated with the deployment of DBSs, focusing on the aspects that are both fundamental and unique to their operation. First, the inherent mobility of DBSs introduces additional degrees of freedom in the spatial domain, enabling flexible placement across a broader range of 3D positions. This flexibility significantly enhances the potential for optimized network configurations. Second, the elevated altitude of DBSs allows them to bypass terrestrial obstacles, making them well-suited for long-range backhaul technologies such as free-space optics (FSO) and mmWave, which can be used to establish links with other DBSs and terrestrial terrestrial macro base stations (MBSs). Building on these capabilities, this thesis demonstrates that **the combination of DBS mobility and altitude with advanced AI-based optimization techniques enables the realization of a highly ubiquitous and adaptable wireless system that ensures seamless coverage across broad service areas while dynamically responding to the changing demands of GNs**.

The main contributions of this thesis can be summarized as follows:

1. **Chapter 1** provides a comprehensive characterization of DBS challenges and opportunities. Specifically, UAVs are categorized based on various aspects such as altitude,

size, range, and configuration, which helps identify their roles as DBSs. The chapter also discusses regulatory and architectural challenges specific to DBS deployment. Finally, it defines the research scope by focusing on DBS-specific challenges such as 3D placement, energy limitations, and backhaul provisioning, while excluding well-established MAC-layer problems.

2. **Chapter 2** introduces the simulation models and scenarios used throughout the thesis. It includes various GNs distribution and mobility models, urban topology modeling, fronthaul and backhaul channel models, and energy consumption and harvesting models. This chapter is a foundational reference for simulating diverse wireless networks under different assumptions and deployment scales.

3. **Chapter 3** proposes innovative fronthaul-aware placement strategies that optimize DBS positions based solely on the GN-to-DBS channel performance. Two scenarios are considered:

   (a) **Known GNs locations:** A modified expectation-maximization (EM) clustering algorithm is proposed using signal-to-interference-plus-noise ratio (SINR)-based similarity metrics to ensure coverage and reduce interference.

   (b) **Unknown GNs locations:** A model-free, multi-agent Q-learning algorithm is developed for autonomous DBS placement based on historical SINR reports from GNs, achieving improved energy efficiency, fairness, and interference mitigation.

   The chapter then proceeds by introducing a novel model to calculate the access probability of vehicular users covered with a DBS using beamforming. This metric is used in an mixed-integer linear program (MILP) formulation to determine the placement of DBSs and their beams footprints, while minimizing the number of active DBSs.

4. **Chapter 4** presents a novel algorithm for optimizing the transmission power settings of DBSs, based on Monte Carlo Tree Search. This chapter shows that power optimization can outperform height tuning regarding energy efficiency and link reliability. The proposed method also supports dynamic DBS deactivation by assigning zero transmission power to redundant nodes.

5. **Chapter 5** focuses on new strategies for backhaul-aware drone placement. The problem is formulated using visibility graphs to satisfy LOS constraints in urban environments. The proposed placement framework is extended to account for solar energy harvesting using on-board PV panels and to incorporate reconfigurable intelligent surfaces (RISs) to improve connectivity in obstructed areas.

6. **Chapter 6** addresses the challenge of finding an innovative method of joint fronthaul-backhaul placement. A modified hierarchical clustering (HC) algorithm is proposed to simultaneously satisfy GN coverage under a minimum signal-to-noise ratio (SNR) constraint and ensure backhaul connectivity through an N_B-neighbors constraint. The solution achieves this without increasing the number of required DBSs compared to conventional methods.

7. **Chapter 7** introduces a novel multi-hop wireless backhaul mesh design framework for DBSs. The drone network problem (DNP) is formulated as a mixed-integer linear program under flow and degree constraints. A tailored genetic algorithm (GA) is proposed to solve this problem while maximizing redundant throughput. The chapter concludes by integrating the placement with HC and backhaul design with GA components into a unified adaptive framework that dynamically adjusts to network changes.

In summary, this thesis provides a complete methodology for designing next-generation airborne networks. The proposed system integrates AI-based placement, transmission power control, energy harvesting, and mesh routing to address the unique challenges of DBS-assisted 3D wireless networks.

Author's Published Contributions

The contributions of the author of this thesis have been published in the following journals and conference papers:

1. Hoffmann, M., Janji, S., Samorzewski, A., Kułacz, Ł., Adamczyk, C., Dryjański, M., Kryszkiewicz, P., Kliks, A. & Bogucka, H. *Open RAN xApps Design and Evaluation: Lessons Learnt and Identified Challenges*. IEEE Journal on Selected Areas in Communications, vol. 42, no. 2, pp. 473–486, 2024.

2. Sroka, P., Kułacz, Ł., Janji, S., Dryjański, M. & Kliks, A. *Policy-Based Traffic Steering and Load Balancing in O-RAN-Based Vehicle-to-Network Communications*. IEEE Transactions on Vehicular Technology, vol. 73, no. 7, pp. 9356–9369, 2024.

3. Janji, S. *Wykorzystanie Powietrznych Przełączalnych Inteligentnych Powierzchni Do Komunikacji Międzypojazdowej*. Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, vol. 1, pp. 358–361, Aug. 2024.
https://sigma-not.pl/publikacja-150619-2024-4.html

4. Janji, S., Sroka, P. & Kliks, A. *Enhancing V2X Communications with UAV-mounted Reconfigurable Intelligent Surfaces*. In: Proc. 2024 IEEE PerCom Workshops, pp. 708–713, 2024.

5. Janji, S., Wawrzyniak, P., Formanowicz, P. & Kliks, A. *Integrating UAV-Enabled Base Stations in 3D Networks: QoS-Aware Joint Fronthaul and Backhaul Design*. arXiv preprint, 2024. https://arxiv.org/abs/2404.17547 [submitted to IEEE Internet of Things Journal]

6. Janji, S. & Kliks, A. *Multi-Agent Q-Learning for Drone Base Stations*. In: Proc. 2023 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 261–266, 2023.

7. Janji, S. & Kliks, A. *Drone Base Stations Transmission Power Control and Localization*. In: Mobile and Ubiquitous Systems: Computing, Networking and Services,

LNICS, vol. 492, Springer, pp. 356–377, 2023. https://link.springer.com/10.1007/978-3-031-34776-4_19

8. Janji, S. & Sroka, P. *RIS-aided Multi-hop Backhauling for 5G/6G UAV-assisted Access Points*. Journal of Telecommunications and Information Technology, vol. 2, pp. 63–69, Jun. 2023. https://jtit.pl/jtit/article/view/1190

9. Janji, S. & Sroka, P. *Wykorzystanie Rekonfigurowalnych Inteligentnych Matryc Antenowych w Łączu Dosyłowym Sieci 5G/6G Wykorzystującej Bezzałogowe Statki Powietrzne*. Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, vol. 1, pp. 87–90, Aug. 2023. https://sigma-not.pl/publikacja-144743-2023-4.html

10. Janji, S. & Kliks, A. *Neural Networks for Path Planning*. arXiv preprint, 2022. https://arxiv.org/abs/2207.00874

11. Janji, S., Samorzewski, A., and Kliks, A. *Procedura Rozmieszczenia Mobilnych Przekaźników Dosyłowych FSO Zasilanych z OZE*. Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, vol. 1, pp. 290–293, Aug. 2022. https://sigma-not.pl/publikacja-138994-2022-4.html

12. Janji, S., Samorzewski, A., Wasilewska, M. & Kliks, A. *On the Placement and Sustainability of Drone FSO Backhaul Relays*. IEEE Wireless Communications Letters, vol. 11, no. 8, pp. 1723–1727, 2022.

13. Janji, S. & Kliks, A. *Energy-Efficient User Clustering for UAV-Enabled Wireless Networks Using EM Algorithm*. In: Proc. 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6, 2021.

Chapter 1

Introduction

This chapter introduces the foundational concepts of DBSs in wireless networks, highlighting their classifications, regulatory constraints, and roles in enhancing network performance. Various UAV types are listed based on altitude, size, configuration, and operational range, followed by a discussion on current aviation regulations, particularly those relevant to European deployment scenarios. The chapter then outlines the unique opportunities and challenges of integrating DBSs into wireless systems, including spatial placement, energy limitations, backhaul provisioning, and interference management. Finally, the research scope and methodology adopted in this thesis are presented.

1.1 UAV Classifications

UAVs can be classified according to various features that define their design and operational characteristics. These classifications help identify a UAV's suitability for specific applications in wireless networks. Key criteria include operational altitude, aerodynamic configuration, size, payload capacity, and mission range [2, 4]. Below, several commonly used classification schemes are discussed. Fig. 1.1 summarizes major classifications.

Operational Altitude

Based on operational altitude, drones are categorized as either low altitude platforms (LAPs) or high altitude platforms (HAPs) [5, 2, 4]. LAPs operate at altitudes of up to 10 km. They are generally more affordable and quicker to deploy, making them suitable for providing coverage to small or localized areas. However, their typical flight duration is limited to approximately 40 minutes, which restricts their use in long-term missions.

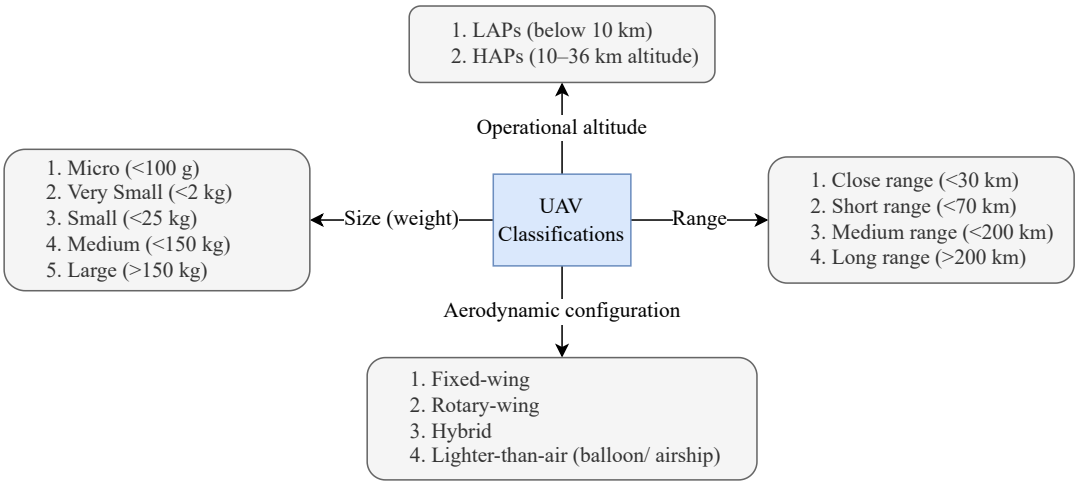


Figure 1.1 Categories of UAVs concerning operational altitude, size, range, and aerodynamic configuration.

In contrast, HAPs operate in the stratosphere at up to 36 km altitudes. They can cover much larger areas and offer extended flight durations of up to 100 days. Additionally, they are capable of carrying heavier payloads. These advantages come at the expense of increased deployment cost and system complexity.

Aerodynamic Configuration

UAVs can also be classified according to their aerodynamic configuration, which includes fixed-wing, rotary-wing, hybrid platforms, and lighter-than-air systems such as balloons and airships [4, 2]. Fixed-wing drones typically provide longer flight durations and lower power consumption due to their efficient aerodynamics, but they cannot hover. On the other hand, rotary-wing UAVs can remain stationary in the air and perform vertical take-off and landing at the cost of higher energy consumption. Lighter-than-air platforms, which usually fall under the HAP category, can support larger payloads and offer extended endurance, making them suitable for high-altitude, long-duration missions.

Size

UAVs may also be classified by size, typically defined by their total weight, including the heaviest possible payload. In [2], five size categories are proposed: Micro (<100 g), Very Small (<2 kg), Small (<25 kg), Medium (<150 kg), and Large (>150 kg). Drones in the Micro category are generally limited to carrying lightweight sensors for wireless data collection.

Very Small drones may support basic communications equipment, such as a cellular user equipment (UE), while Small and Medium UAVs can accommodate lightweight airborne BSs. Large UAVs are generally used for complex or military applications that require high endurance and heavy payload support.

Alternatively, the European Union Aviation Safety Agency (EASA) categorizes drones under the 'open' (i.e., low risk) category into four different weight categories [6]:

1. C0: under 250 g
2. C1: under 900 g
3. C2: under 4 kg
4. C3 and C4: under 25 kg

Range

Range, defined as the maximum controllable distance from a ground station or controller, can also serve as a classification criterion. It can vary from tens of meters to several hundred kilometers, depending on the communication technology and platform design. In the case of HAPs connected via satellite links, the range becomes virtually unlimited [5]. In [7], UAVs are categorized into four range classes: close range (<30 km), short range (<70 km), medium range (<200 km), and long range (>200 km).

Payload

The payload capacity of a UAV is another key classification parameter. Commercial UAVs can carry payloads ranging from just a few kilograms in compact models to as much as 500 kg in larger airship-based platforms [4]. Payload capacity influences the range of onboard equipment deployed, such as sensors, communication relays, or edge computing servers.

1.2 Regulations

As with any emerging technology, regulations play a crucial role in guiding both the development and deployment of UAVs. Governmental agencies have introduced rules categorized according to different use cases, addressing privacy, public safety, security, and collision avoidance concerns. In Europe, the authority responsible for regulating UAV operations is the EASA. In 2019, two key regulations—2019/947 and 2019/945—were introduced to define the rules and procedures for UAV operations [8]. These rules follow a risk-based

approach, focusing not on the drone's purpose (leisure or commercial), but on its technical characteristics and the nature of the operation.

Three risk-based operational categories are defined:

1. **Open category:** For low-risk operations that do not require prior authorization and are conducted within visual line of sight (VLOS). Drones in this category must weigh less than 25 kg and avoid flying over assemblies of people. Strict limitations on altitude and proximity to people apply. DBSs that comply with these constraints may be operated under this category.

2. **Specific category:** For medium-risk operations that require prior authorization. These typically involve flights beyond VLOS, operations over populated areas, or drones exceeding 25 kg in weight. This category is particularly relevant for DBSs intended for broader coverage or operation in urban environments.

3. **Certified category:** The highest-risk category, intended for operations similar to manned aviation. It includes carriage of passengers or dangerous goods and requires certification of the UAV, the operator, and possibly the remote pilot.

To operate DBSs, which will typically fall under the specific category, operators must submit a comprehensive concept of operations (ConOps) and a detailed risk assessment. Other regulatory considerations include the maximum take-off mass (MTOM), communication systems, and control redundancy.

Altitude limitations are also significant for this thesis, as altitude is a critical factor in DBS placement for optimizing network performance. Most UAV operations are limited to a maximum altitude of 120 meters. Exceptions include operations near or above structures taller than 105 meters, where drones may fly up to 15 meters above the structure, and authorized missions under the specific category, which may allow flight up to 150 meters, subject to certain conditions. Operations at even higher altitudes require further authorization and coordination with local aviation authorities.

1.3 The Role of Drones in Wireless Connectivity

The deployment of DBSs has been extensively studied in recent years [5, 9, 1, 10, 11, 2], and a wide spectrum of use cases has been identified. Drones offer a wide range of use cases in the future network. They provide connectivity to GNs, act as relays between cells (including terrestrial and non-terrestrial stations) and/or other cells or GNs. They can also relay data between GNs in device-to-device communications (D2D) communications. Another

popular application that has emerged is their function as data collectors for supporting sensor networks, increasing the energy efficiency of these networks by designing flight trajectories that bring UAVs close to sensors during collection. They can also be used as caching servers to improve the efficiency of networks. Finally, they are also considered mobile edge computing servers deployed for offloading computational load from devices. In this thesis, drones are considered as BSs providing connectivity to GNs (Chapters 3, 4, 6) or as relay nodes transmitting backhaul data between cells (Chapters 5, 6, 7).

1.4 Challenges in Deploying DBS

DBSs can be deployed as standalone providers of connectivity for GNs, or in a heterogeneous manner as cells integrated in a larger network comprised of satellite nodes and terrestrial BSs [5]. Each node type offers distinct advantages and disadvantages. While a satellite station is characterized by ubiquitous connectivity, it also introduces high latency and low data rates. Terrestrial BSs on the other hand provide greater rates and low delay, but are limited in terms of coverage and susceptible to disruption by natural disasters. DBSs, with their flexibility and ease of deployment, can be deployed as a cost-effective solution to provide LOS coverage and dynamically adapt their locations according to the needs on the ground. Compared to satellite BSs, they provide a more cost-effective solution and reduced latency. However, a UAV is limited by its energy and is vulnerable to extreme weather conditions. Also, DBSs networks are affected by increased interference, and, so far, they suffer from stringent regulatory and safety issues.

The problem of DBSs deployment presents its own unique challenges to the problem of wireless system design in addition to the typical challenges faced with terrestrial networks. Specifically, when deploying DBSs we have to take into account the following factors.

Placement (space)

The implications that DBSs present to the performance and optimization of wireless networks in the space dimension due to their mobility are the most defining and unique factor of DBS-assisted networks. This mobility enables dynamic adjustments in network topology where DBSs are strategically repositioned to improve the network's performance. Nevertheless, this presents new factors that should be considered when designing such wireless networks [12]. Deciding the placement of DBSs should consider the following aspects:

- Avoiding collision with obstacles and other DBSs.
- Reducing interference to other nodes.

- Minimizing path-loss to served GNs.
- Proximity to backhaul gateway nodes.
- Proximity to base or charging stations.
- Energy-related factors such as recharging strategies, reducing aerodynamic and transmission energy consumption, proximity to energy harvesting locations, and maximizing the energy gained from different sources.
- Reducing the total number of DBSs deployed, reflecting on the system's energy efficiency and cost effectiveness.
- Adhering to local regulations.

Backhaul

DBSs are typically limited to wireless backhaul mechanisms. In comparison to their terrestrial counterparts, and due to their altitude, DBSs can leverage directive propagation schemes such as FSO and mmWave that offer higher data rates, less interference, and longer transmission range when compared to Wi-Fi and cellular networks that operate on lower frequencies. Nevertheless, the design of DBSs' backhaul should address the following challenges [13]:

- Ensuring adequate capacity to handle the data rates of GNs and control signalling of the system.
- Minimizing latency to improve real-time data transmission.
- Maintaining reliable transmission and robustness against failures, including re-establishing connectivity paths in the event of node failure and adapting to varying weather conditions.
- Managing interference either by frequency planning or using directed signal propagation.
- Achieving scalability to accommodate larger networks or traffic efficiently.
- Optimizing energy consumption.
- Securing backhaul links.
- Ensuring cost-effectiveness.
- Compliance with regulations such as those related to spectrum usage.

UAV energy

Energy management is critical in designing solutions involving UAVs due to their limited battery life and the need for prolonged operational periods. When possible, efficient energy replenishment techniques are essential to enhance UAV endurance, allowing for longer flight times and expanded mission capabilities. Here are six potential charging techniques for UAVs, each with its own set of advantages and limitations [14]:

1. **Battery Swapping**: This technique involves replacing a depleted battery with a fully charged one. While it allows for rapid replenishment and minimal downtime, it requires precise alignment and mechanical systems for battery handling, which can be complex and costly [15].

2. **Ground-based Charging Stations**: These stations allow UAVs to land and charge via direct electrical connections. This method is reliable, but limits the UAVs' range to the proximity of available charging stations..

3. **Laser Beam Charging**: UAVs can be charged by lasers directed from the ground, which convert light to electricity using photovoltaic cells. This method can charge UAVs over long distances but poses safety risks and requires clear line-of-sight between the UAV and the laser source [16]

4. **Transmission Line Charging**: This innovative approach involves UAVs drawing power directly from electrical transmission lines, either by landing on them or hovering close. It taps into the electromagnetic fields around power lines, but it is technically challenging and poses significant safety risks. [17, 18]

5. **Wireless Charging**:

   - *Inductive Power Transfer (IPT)*: Utilizes electromagnetic fields generated by coils to transfer energy. It is efficient at short ranges, but its efficiency drops significantly with distance [19, 20].
   - *Resonant Capacitive Wireless Power Transfer (RCWPT)*: Enhances capacitive coupling through resonance, allowing for more efficient power transfer over longer distances than IPT and with less sensitivity to alignment.

6. **Solar Panels on UAVs**: Direct integration of photovoltaic panels (PV) panels on UAVs can harness solar energy during flight, providing a continuous charge. This method depends heavily on weather conditions and typically cannot fully power a UAV but can significantly extend flight times [21–23].

7. **Wind Power Harvesting**: UAVs can harvest wind energy through techniques such as dynamic soaring or gust energy extraction, where they exploit wind gradients or turbulence to gain kinetic energy. While still an emerging area, these methods can potentially reduce propulsion energy consumption and extend endurance without adding significant onboard hardware [24].

In this thesis, option 6. that involves PV panels is considered, as it introduces an interesting placement problem in an urban scenario that is different from the problems posed by other techniques, which usually involve similar trajectory planning approaches [14]. A study is performed on the gain introduced by installing PV panels on top of drone relay stations (DRSs) in terms of prolonging their service life.

Load balancing

Ensuring a uniform distribution of load across the network to prevent specific nodes from being overloaded is a critical aspect of any wireless system design [25]. Such a distribution maximizes the utilization of network resources, thus increasing the system's efficiency. It also reduces the implications of excessive demand, such as call dropping and latency.

Transmission power control

Controlling the transmission power of cells in a cellular network is crucial for optimizing network performance. In addition to its role in load balancing, the management of transmission power helps to ensure Quality of Service (QoS), minimize interference, and maximize energy efficiency.

Mobility handling

Efficiently managing the mobility of GNs and their handovers is crucial for achieving seamless service continuity and supporting services such as autonomous driving and massive IoT deployments.

Integration with terrestrial networks

In the cases where DBSs are deployed contiguously with terrestrial stations, it is important to coordinate the different aspects mentioned previously while considering the terrestrial BSs and controlling their parameters (e.g., power, frequency-time allocation, etc.) as well. This is important to maximize network performance without causing disruption by deploying DBSs.

Frequency-time resource allocation

The critical role of effective frequency-time resource allocation in modern cellular networks is well-documented and widely recognized as fundamental to their design and operational efficiency [26]. The process involves strategically scheduling and distributing frequency bands and time slots to optimize network performance, enhance service quality, and support high-demand applications. While the allocation of resource blocks to GNs within cellular networks has been extensively explored and is highly developed, it remains outside the focus of this thesis, which centers on other DBS-related aspects of network optimization.

1.5 Research Framework and Methodology

1.5.1 Scope

In this thesis, the first five optimization aspects discussed above are considered (i.e., placement, backhaul, energy, load balancing, and transmission power control). It is assumed that issues related to frequency-time resource allocation and mobility have been sufficiently researched and optimized within the literature on terrestrial networks. The principles and techniques developed can be adapted for DBS network management. This assumption allows us to focus DBS-specific research on addressing novel challenges. This approach also includes neglecting aspects related to integrating DBSss' fronthaul radio access network (RAN) access with terrestrial nodes, under the assumption that a DBS will serve only a subset of GNs within the coverage area of terrestrial nodes, determined by frequency-time scheduling mechanisms (i.e., GNs are divided into terrestrial and non-terrestrial subsets with different frequency bands or time slots) or spatial user splitting.

1.5.2 Methodology

To evaluate the proposed algorithms and system design, the author developed a simulation framework capable of emulating various wireless network scenarios. The simulators were designed to support configurable parameters for different network types, deployment environments (urban, suburban, rural), system scales (from local to wide-area coverage), and mobility models (including static, clustered, and dynamic GN distributions). It also emulates different system requirements such as minimum coverage thresholds, backhaul reliability constraints, per-user throughput needs, energy consumption of drones, and energy harvesting through solar panels.

The simulation framework operates primarily in the spatial domain, optimizing the 3D placement of DBSs while assuming full-buffer traffic conditions. Time and frequency-domain aspects, including scheduling, interference coordination, and channel reuse, are abstracted away and left to MAC-layer schedulers in future work. This spatial abstraction enables an analysis focused on DBS-specific factors such as placement, coverage, and connectivity under steady-state demand.

By documenting all simulation assumptions and systematically varying parameters, a comprehensive methodology is provided that can be extended to simulate and test future wireless networks with diverse technologies, user behaviors, and deployment strategies.

Chapter 2

System Model

This chapter introduces the models and assumptions underlying the problems explored in the following chapters. It begins with an overview of user distribution and mobility models that simulate the network environment and evaluate the proposed solutions. Then, the author explains how the model of an urban area topology is obtained, upon which algorithms may depend. Subsequently, the channel models for both fronthaul and backhaul layers are introduced. Finally, in some cases, an energy consumption model for the DBS is needed, which is presented in the subsequent section. Furthermore, the model for energy harvesting through PV panels, which is leveraged in one of the solutions, is presented next.

In the system model, the set of DBSs, denoted by $\mathcal{M}$, consists of M elements. The position of each DBS, j, where j belongs to the set $\{1, 2, \ldots, M\}$, is specified by its coordinates $\boldsymbol{m}_j = [x_j, y_j, h_j]^T \in \mathbb{R}^3$. The GNs are represented by a total count of U, forming the set $\mathcal{U}$. The location of GN n, within the set $\{1, 2, \ldots, U\}$ and assumed to be at ground level, is given by $\boldsymbol{u}_n = [x_n, y_n]^T \in \mathbb{R}^3$. Furthermore, the set $\mathcal{B}$ encompasses B additional stations that do not qualify as DBSs. These include terrestrial MBSs, which may offer fronthaul connectivity to GNs as well as backhaul connections within the network, alongside other aerial or space entities functioning as backhaul gateways for the DBSs. In this context, the analysis primarily focuses on MBSs, defined by their locations $\mathbf{b}_k = [x_k, y_k, h_k]^T \in \mathbb{R}^3$, where k extends from 1 to B.

The three-dimensional (3D) Euclidean distance between points indexed by j and n is given as

$$d(j,n) = \sqrt{d_{2\mathrm{d}}^2(j,n) + (h_j - h_n)^2}, \tag{2.1}$$

where

$$d_{2\mathrm{d}}(j,n) = \sqrt{(x_j - x_n)^2 + (y_j - y_n)^2} \tag{2.2}$$

represents the 2D distance on the ground plane. These formulas calculate distances between various pairs of points, such as two GNs, two BSs, or a GN and a BS. The context will determine the specific type of pair under consideration.

Note that in the formulas presented above and hereafter, the time variable t is not explicitly shown. However, it is assumed that each variable also depends on t as its value changes with time. In certain cases, t will be explicitly highlighted for clarity.

2.1 User Distribution and Mobility Models

The simulated networks are classified according to the considered coverage area into two categories:

1. *large scale networks* where drones and balloons or other high-altitude platforms (HAPs) are deployed over large swathes of land to provide connectivity throughout entire cities. In general, in such cases, UAVs are deployed at high altitudes and with low density.

2. *small-scale networks* in which drones cover a small region with few users. Here, a denser deployment of drones is modeled at a lower altitude to mitigate interference.

Note that the two types of network can overlap and connect, and in both cases, the network will also be connected to other terrestrial or space base stations to provide connectivity to the core network, thus forming a 3D network [27]. These deployments are illustrated in Figure 2.1 together with the different backhaul schemes that can be utilized to interconnect DBSs with each other and with other network elements.

In the case of large-scale networks, in addition to serving GNs that can be moving, a DBS may also provide connectivity to both terrestrial MBSs and SBSs in cases where an alternative backhaul path to these nodes does not exist (e.g., due to a disaster). On the other hand, it is assumed that DBSs are directly serving GNs in case of small-scale deployments.

As expected, the distribution of users in each case is very different. It is safe to assume that across a large area and when UAVs are flying at a high altitude, different GNs such as vehicles, pedestrians, or TBSs have a static deployment since, even though some GNs may be moving, their relative speed and location differences are less emphasized, and to handle this mobility, a network recomputation based on a new distribution snapshot can be performed. Furthermore, the distribution of users in such cases resembles that of cluster processes where GNs are grouped in clusters around TBSs' locations, as observed in real environments [28].

On the other hand, when DBSs are deployed at a low altitude, GNs' movement has a more pronounced effect on the link distance and therefore its gain, and their distribution is more uniform than in the previous scenario.

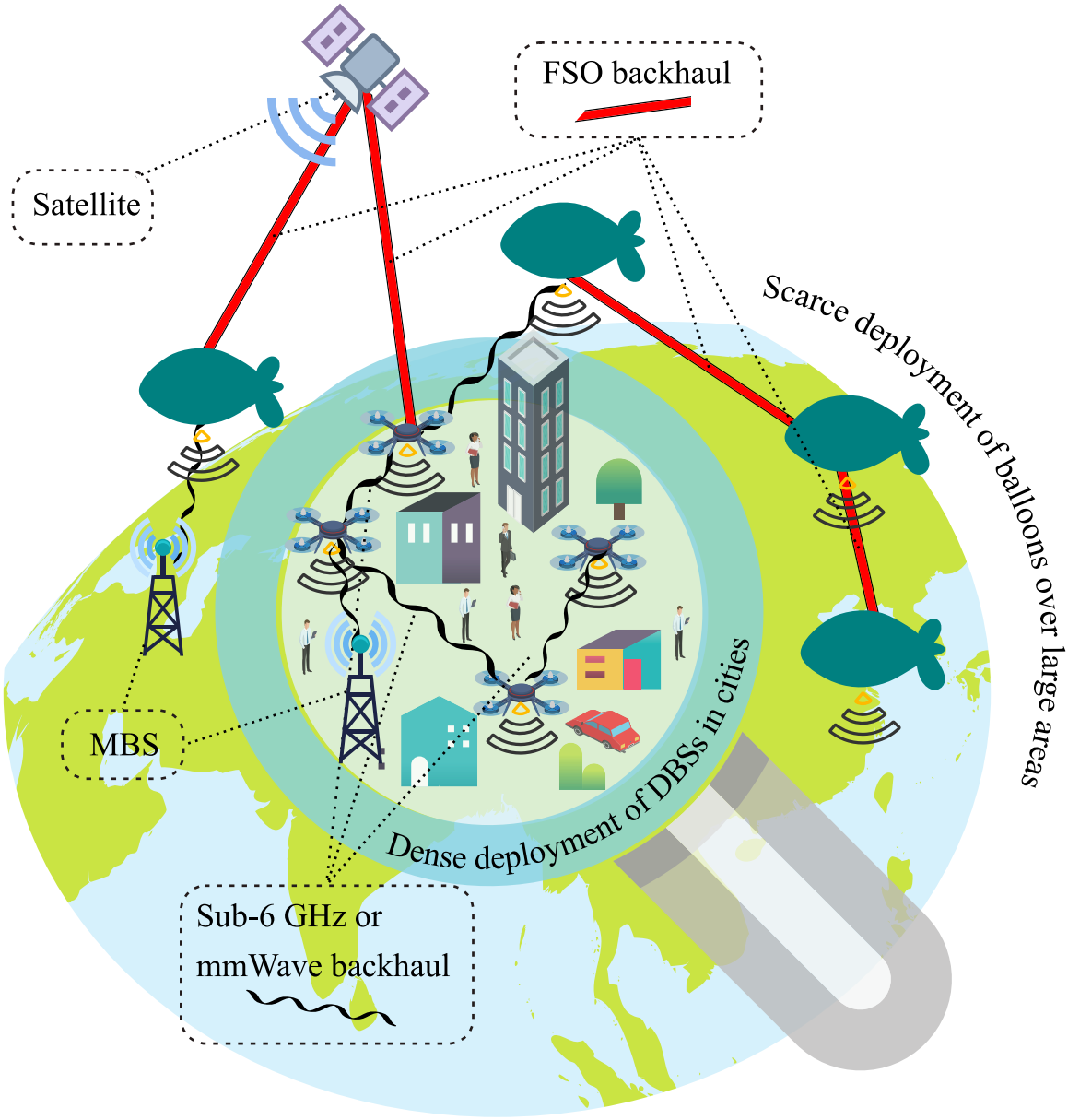


Figure 2.1 Illustration of diverse UAV-based 3D network configurations for B5G and 6G wireless communication. Low-altitude DBSs facilitate dense network deployments in urban settings to enhance spectrum reuse and LOS connectivity. High-altitude platforms, such as balloons, extend coverage across vast areas. Backhaul solutions include inter-UAV links and connections to terrestrial base stations or satellites, utilizing FSO, cmWave, and mmWave technologies for robust wireless communication.

The following subsections introduce the user distribution and mobility used in different parts of the thesis.

2.1.1 Static users

When the GNs are not moving, one of the following two distribution models is used.

Uniform distribution of static users

This is the simplest approach in which a given number of users, U, are distributed uniformly throughout the area, which does not resemble a realistic network topology. A similarly simple approach is to model GNs with a Poisson point process (PPP), which still does not reflect real scenarios where users are usually clustered [29].

Clustered distribution of static users [30]

In real-life scenarios, user presence and movement are shaped by physical characteristics (parks, buildings, etc.) and social factors (events, attractions, work locations, etc.), making a more accurate representation of their locations possible through different cluster processes. The Thomas cluster process (TCP), which is a type of Poisson cluster process (PCP), as detailed in [28, 30], is one such model. Here, cluster centers are placed according to a PPP across the area with intensity λ_{TCP}. GNs are then distributed around the center of cluster i according to a normal distribution with mean $\boldsymbol{\mu}_i \in \mathbb{R}^2$, corresponding to the location of the cluster center and variance $\sigma_i^2 \in \mathbb{R}$ in each dimension. The probability of a sample location is obtained using the multivariate Gaussian probability density function (PDF) as

$$\mathcal{N}(\boldsymbol{u}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) = \frac{1}{\sqrt{(2\pi)^2 |\boldsymbol{\Sigma}_i|}} \exp\left(-\frac{1}{2}(\boldsymbol{u} - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{u} - \boldsymbol{\mu}_i)\right), \tag{2.3}$$

where $\boldsymbol{u} \in \mathbb{R}^2$ is the RV for the GN location, and $\boldsymbol{\Sigma}_i = \begin{bmatrix} \sigma_i & 0 \\ 0 & \sigma_i \end{bmatrix}$ is the covariance matrix.

The procedure for generating GN location data in an area of A square meters is outlined below, with Figure 2.2 providing a visual representation of a GN distribution generated through the TCP method.

1. Generate $N_{\text{TCP}} = A\lambda_{\text{TCP}}$ cluster locations.

2. For each cluster, generate a random number of users according to a Poisson distribution with intensity λ_{U}.

3. For each child (GN) in a cluster, generate their location based on (2.3).

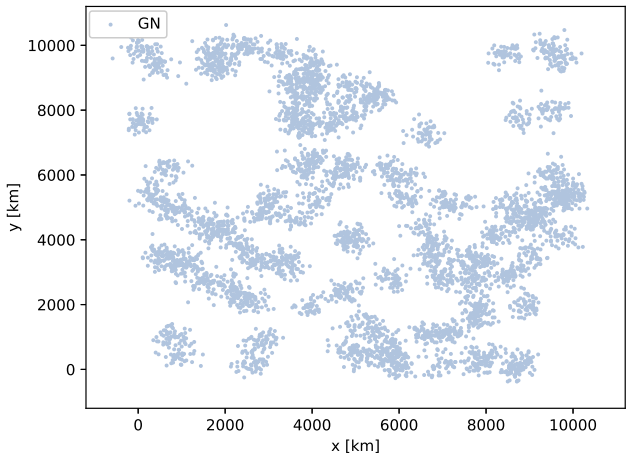


Figure 2.2 Example of GNs generated according to a TCP in an area of $A = 10 \times 10$ km, with $\lambda_{\text{TCP}} = 10^{-6}$, $\lambda_{\text{U}} = 50$, and $\sigma_i^2 = 200$ for all clusters. This approach yields more realistic results than a uniform distribution of GNs.

2.1.2 Moving users - mobility models

In simulations where GNs are moving, one of the two following models is used. The first one, random waypoint mobility model (RWP), is used when there is no information about the topology of buildings in the simulation area; otherwise, the second model, obstacle mobility model (OMM), is used when the buildings' edges are defined.

Random Waypoint (RWP) Model [31]

One of the simplest and oldest approaches to simulate user movement is the RWP method [32], where

1. each GN selects a random destination to move to at a randomly selected speed.
2. When reaching its destination, the GN may pause for a randomly selected duration,
3. and then selects a new destination, thus repeating the cycle.

Although this model effectively introduces dynamicity into the network topology for simulation, a further improvement can be made by introducing dependency between movements of different clusters of GNs, which is achieved by the RWP model [31]. In this model, groups

of GNs are simulated in which each user stays in proximity to its group center throughout the simulation. The group center location varies along with the movement of its *leader*, which is a GN that can be simulated according to the standard RWP approach. The group members' locations are first initialized around the leaders at the starting time step. Then, to preserve their proximity, they move in the following manner:

1. When a leader selects a new destination, the path towards that endpoint is divided into checkpoints.

2. For each group member, the next destination is selected as a random location within a certain radius of the next checkpoint, thus introducing randomness in both directions and velocities of members.

This simple approach can approximate different realistic scenarios where users move in groups such as large-scale disaster recovery with different responder groups working together, events with groups of attendants, battlefield with battalions moving together, etc.

Despite its wide usage, the RWP (and the RWP) are rather unrealistic for several reasons. One well-documented issue is the phenomenon known as the *speed decay* problem where GNs average velocities decrease as they converge towards the center of the simulation area, thus affecting the simulation results in a non-realistic way with larger node density near the center. Other shortcomings include sharp turning near boundaries and sudden stopping of nodes that are different from actual patterns of users. In real life, mobility patterns involve predictable patterns that align with specific paths within a particular geographic area. To overcome these issues, when the topology of the building environment is known, the following mobility model is used.

Obstacle Mobility Model (OMM) ([33]

With advancements in geographic information system (GIS) and the availability of open-source platforms like OpenStreetMap (OSM) [34], obtaining detailed topologies of urban areas, including building structures and various obstacles, has become significantly easier than in the past. This accessibility allows for more accurate and comprehensive modeling of urban environments, enhancing simulations in network planning, urban development, and emergency response scenarios. The detailed data from OSM, for instance, provides a rich source of information on urban layouts, buildings' shapes, sizes, and locations, facilitating more realistic and effective planning and analysis. Building on the understanding that people usually follow certain paths that depend on surrounding obstacles within an environment, such as pathways in campuses or streets in cities, the authors in [33] introduced an approach

where GNs movement lanes are extracted from the geometric description of surrounding obstacles.

The lanes are derived from a Voronoi diagram created using building corners as vertices. This diagram segments the plane into regions, or Voronoi cells, ensuring that each point within a region is nearest to its corresponding vertex. These boundary lines, equidistant from adjacent building corners, represent user movement pathways, mirroring the realistic behavior of navigating mid-way between buildings. Figure 2.3 shows an example of such paths generated for the Madrid grid simulation environment [35], which is used in different parts of this thesis.

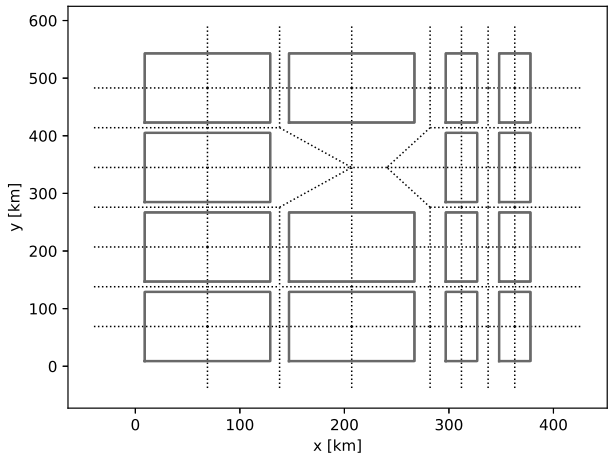


Figure 2.3 GN movement paths in the Madrid grid simulation environment [35], derived from a Voronoi diagram of building corners, as per the OMM model [33].

After obtaining the paths that are defined with their respective edges and vertices, GNs locations are selected randomly from the locations of the Voronoi graph vertices. Subsequently,

1. Each GN randomly selects another vertex in the graph as the next destination to travel to, and,
2. through employing Dijkstra's shortest path algorithm, the path for reaching a destination is determined.
3. GNs traverse their paths with different speeds until reaching their destination.
4. Then, upon reaching its destination, each GN pauses for a random duration before repeating the cycle.

The authors show by simulations that this mobility model generates more realistic results than, e.g., the RWP model, which modifies the network's performance substantially.

2.2 Fronthaul Channel Modeling

In the study of UAV-GN wireless communication channels, the literature presents a wealth of models that cater to various environments, ranging from urban to rural settings [36, 37]. These models may include phenomena that influence signal behavior in distinct manners, in addition to the fundamental aspect of the mean PL, which encapsulates the average signal attenuation experienced over distance. However, this metric does not factor in the rapid signal intensity variations known as fading.

Small-scale fading arises primarily due to multipath propagation, where the signal traverses multiple paths before reaching the receiver. The converging signals, each with unique phases and amplitudes, can amplify or cancel each other out based on their phase alignments. This phenomenon can manifest uniformly across the channel's bandwidth or vary, leading to frequency-selective fading. Additionally, models may account for variations related to shadowing, where obstacles obstruct the signal's path, and Doppler spread, a fading type induced by the relative motion between the transmitter and receiver or the movement of surrounding objects, which may even cause frequency shifts.

In this thesis, the focus narrows to the mean path loss (PL) in most cases, setting aside the variable nature of fading phenomena and only including its average behavior. This simplified approach is sufficient for the exploration, design, and evaluation of DBSs deployment strategies. By prioritizing the optimization of average received power across the space domain, along with interference in some cases, the author aims for an overall enhancement in network performance. The complexities of fading related to multipath propagation and Doppler spread are delegated to network management and optimization processes. Specifically, the scheduling algorithms in modern networks are designed to allocate time-frequency resources dynamically. Through intelligent scheduling, the network can effectively mitigate the adverse impacts of fading, ensuring reliable communication even in the face of these challenging conditions. This deliberate separation allows for a more focused analysis on spatial deployment strategies, under the assumption that the network's scheduling mechanisms adeptly manage temporal and frequency domain challenges.

2.2.1 DBS-GN channel model

To model the DBS-GN channel, the widely used simplified PL model introduced in [38] is used in this thesis. The authors of this model account for additional losses—such as shadowing and scattering beyond free-space loss—by averaging these effects into a single parameter, denoted by η. The model differentiates between LOS and non-line of sight (NLOS) propagation conditions. For a DBS j and GN n, the probability of each condition is derived as follows:

$$\mathrm{Pr}_{\mathrm{LOS}}(j,n) = \frac{1}{1+\alpha\exp\left(-\beta(\omega-\alpha)\right)}, \tag{2.4}$$

where α and β are environmental constants and $\omega = \arctan\left(\frac{h_j}{d_{2\mathrm{d}}(j,n)}\right)$ represents the elevation angle to the DBS. The free-space path loss (FSPL) in dB is given by:

$$\mathrm{FSPL} = 20\log\left(\frac{4\pi f_c d(j,n)}{c}\right), \tag{2.5}$$

where f_c is the carrier frequency and c is the speed of light. By adopting fixed additional losses η_{LOS} and η_{NLOS} for LOS and NLOS conditions, respectively, the expected PL value is:

$$L^{[\mathrm{dB}]}(j,n) = \mathrm{FSPL} + \mathrm{Pr}_{\mathrm{LOS}}(j,n)\eta_{\mathrm{LOS}} + (1-\mathrm{Pr}_{\mathrm{LOS}}(j,n))\eta_{\mathrm{NLOS}}. \tag{2.6}$$

Also, the transmission powers of deployed DBSs are represented by

$$\boldsymbol{P} = [P_T(1),\ldots,P_T(M)], \tag{2.7}$$

and that the receiver's noise power is given by σ_0^2, which is also the variance of the additive white Gaussian noise (AWGN). In many cases, and in accordance with with the intention to isolate the framework from frequency-time allocations, the author assumes that each GN n is assigned a bandwidth of $B_n = B_0$ in Hertz. This enables understanding the behavior of algorithms under equal conditions for all users and focusing on their performance. Therefore, given the total bandwidth of a DBS, B_D, the power received by GN n from the tranmission of DBS j can be calculated as

$$P_R(j,n) = \frac{B_n P_T(j) 10^{\frac{-L^{[\mathrm{dB}]}(j,n)}{10}}}{B_D}. \tag{2.8}$$

Then, the SNR value at the downlink side is obtained as

$$\mathrm{SNR}(j,n) = \frac{P_R(j,n)}{\sigma_0^2}, \tag{2.9}$$

and the SINR value is

$$\Gamma(j,n) = \frac{P_R(j,n)}{\sigma_0^2 + \sum_{i \neq j}^{M} P_{R(i,n)}}. \tag{2.10}$$

As a result, assuming an allocated bandwidth of B_n, the throughput in bits per second (bps) can be obtained by using the theoretical capacity limit provided by Shannon's formula as

$$R(j,n) = B_n \log_2(1 + \Gamma(j,n)). \tag{2.11}$$

In the works presented below, a max-SINR association scheme is employed between GNs and DBSs which is mapped by the binary variable $W(j,n) \in \mathbb{B}$ defined as

$$W(j,n) = \begin{cases} 1 & \text{if } j = \underset{j'}{\operatorname{argmax}}\, \Gamma(j',n) \\ 0 & \text{otherwise.} \end{cases} \tag{2.12}$$

Additionally, the author assumes a minimum SINR threshold, Γ_{th}, that each GN must achieve to be considered *served* (i.e., not in an outage state). Consequently, at each time instance, the number of GNs served by a DBS j is

$$N_j = \sum_{n=1}^{U} W(j,n) H_H\left(\Gamma(j,n) - \Gamma_{\text{th}}\right), \tag{2.13}$$

where $H_H(x)$ is the Heaviside step function. Similarly, the total number of served users is given by

$$N = \sum_{j=1}^{M} N_j. \tag{2.14}$$

Environment Variables α and β

Referencing [38], the clear LOS path probability in (2.4) is inspired by the International Telecommunication Union (ITU) method for calculating geometrical LOS probability between terrestrial transceivers as detailed in [39]. The method utilizes geometrical parameters of urban structures and requires three inputs:

1. α_U: the proportion of the built-up area to the total area,
2. β_U: the average number of buildings per square kilometer,
3. γ_U: a scale factor defining building heights, modeled by a Rayleigh distribution $f_R(H) = \frac{H}{\gamma^2} \exp\left(-\frac{H^2}{2\gamma^2}\right)$ where H is the height.

With these parameters defined by the urban environment's geometry, α and β are calculated using the surface polynomial fitting described by equation ([38, Eq. (6)])

$$z = \sum_{j=0}^{3} \sum_{i=0}^{3-j} C_{ij} (\alpha_U \beta_U)^i \gamma_U^j, \tag{2.15}$$

where the coefficients for α and β can be found in Tables 2.1 and 2.2.

Therefore, whenever an urban environment is simulated with a known geometrical description of the buildings, then the model parameters can be adapted as explained above.

Table 2.1 Surface Polynomial Coefficients for α

C_{ij}	$i=0$	$i=1$	$i=2$	$i=3$
$j=0$	9.34×10^{-1}	2.3×10^{-1}	-2.25×10^{-3}	1.86×10^{-5}
$j=1$	1.97×10^{-2}	2.44×10^{-3}	6.58×10^{-6}	-
$j=2$	-1.24×10^{-4}	-3.34×10^{-6}	-	-
$j=3$	2.73×10^{-7}	-	-	-

Table 2.2 Surface Polynomial Coefficients for β

C_{ij}	$i=0$	$i=1$	$i=2$	$i=3$
$j=0$	1.17	-7.56×10^{-2}	1.98×10^{-3}	-1.78×10^{-5}
$j=1$	-5.79×10^{-3}	1.81×10^{-4}	-1.65×10^{-6}	-
$j=2$	1.73×10^{-5}	-2.02×10^{-7}	-	-
$j=3$	-2×10^{-8}	-	-	-

System's Energy Efficiency (EE)

The energy efficiency (EE), measured in bits per joule, is defined below according to [40]. This metric considers the throughput achievable by GNs, which depends on the received SINR, characterized by the signal power from the serving DBSs and interfering DBSs. Assuming that each GN is assinged a uniform bandwidth of B_n, and after obtaining the rates achieved by GNs using (2.11) under a max-SINR association scheme, the EE is computed as

$$\zeta = \frac{R_{\text{total}}}{P_{\text{total}}}, \tag{2.16}$$

where R_{total} is the total throughput achieved by all GNs calculated using (see (2.11) and (2.12))

$$R_{\text{total}} = \sum_{j=1}^{M} \sum_{n=1}^{U} W(j,n) R(j,n), \tag{2.17}$$

and P_{total} represents the total transmission power expended by the DBSs, given by

$$P_{\text{total}} = \sum_{j=1}^{M} P_T(j). \tag{2.18}$$

2.3 Backhaul Channel Modeling

Backhaul connections link cell sites to the main network and can use various technologies, including microwave, mmWave, FSO, and fiber optic cables. Microwave is a popular choice due to its low cost and moderate range, but it's not suitable for DBSs due to its limited range. Fiber cables, while offering high speeds and reliability, are impractical for DBSs because they restrict the mobility of drones.

For DBSs, the focus is on mmWave and FSO technologies in this thesis. mmWave communication provides very high data rates, essential in densely populated urban areas with high demand. However, its effective range is short and can be affected by atmospheric conditions like rain or fog.

FSO technology uses light to transmit data through the air, offering extremely high data rates similar to fiber optics but without the physical constraints. FSO is not affected by radio frequency interference, making it highly secure and reliable, although it is sensitive to weather conditions such as fog and rain, which can disrupt the light signals.

Both mmWave and FSO technologies benefit from using directional communication, which not only enhances the efficiency of the data transfer but also increases security. The direct beams make it difficult for unintended receivers to intercept the signal, providing an added layer of security for the transmitted data. Furthermore, these two technologies can be combined to build a robust hybrid backhaul scheme which avoids the weather issues since each technology is affected by different conditions [13, 41]. By focusing on these wireless technologies, DBSs can maintain their mobility and adaptability, crucial for extending network coverage effectively.

Below, the channel models used for simulating each technology are presented. Note that a backhaul connection can be established between any two nodes from the composite set

$$\mathcal{V} = \mathcal{B} \cup \mathcal{M} = \{\mathcal{B}_1, ..., \mathcal{B}_B, \mathcal{M}_1, .., \mathcal{M}_M\} \tag{2.19}$$

which contains a total of $B+M$ elements. Moreover, the distance in meters for the link between BSs i and l (with indices $i, l \in \{1, ..., B+M\}$) is denoted as $d(i, l)$.

2.3.1 mmWave Channel Modeling

To model the mmWave channel between those BSs assuming clear LOS, a model similar to FSPL is used, which calculates the PL in dB as

$$L_{\text{mm}}^{[\text{dB}]}(i,l) = \text{FSPL}_{\text{d}_0} + 10\alpha_{\text{mm}} \cdot \log_{10}\left(d(i,l)\right), \tag{2.20}$$

where FSPL_{d_0} is the free-space PL at distance $\text{d}_0 = 5$ m and α_{mm} is the PL exponent which is taken to be equal to 2.13 for 38 GHz transmissions [42].

2.3.2 FSO Channel Modeling

The author leverages the model presented in [43], in which the FSO channel gain is given by

$$h_{\text{FSO}} = \eta_{\text{FSO}} h_p h_a h_g, \tag{2.21}$$

where η_{FSO} is the photo-detector responsivity and h_p, h_a, and h_g are the atmospheric loss, atmospheric turbulence induced fading, and geometric and misalignment losses (GML), respectively. The atmospheric loss, h_p, can be expressed as

$$h_p = 10^{-\kappa \frac{d}{10}}, \tag{2.22}$$

where BS indices are omitted in equations (2.22)-(2.26) for simplicity. The coefficient κ $[m^{-1}]$ represents a weather-dependent attenuation coefficient, accounting for various weather conditions, including clear air and varying fog intensities. Concerning the atmospheric turbulence, h_a, which is induced by temperature and atmospheric pressure inconsistencies across the link, the authors in [43] utilize its expected value, assuming its impact is insubstantial relative to the GML. This can be ensured by employing adaptive optics (AO) technology that usually leverages deformable mirrors in the FSO receivers, which is proven to mitigate atmospheric-turbulence-induced losses [44]. Then, adopting the same assumption implies $h_a = \mathbb{E}\{h_a\} = 1$. Regarding the GML, the authors presuppose a link distance spanning several hundred meters, sufficiently exceeding the FSO beam footprint misalignment (i.e., the distance between the incident beam's center on the receiver's lens and the lens's mid-point). This assumption applies to the considered system, where the link distance may even extend into the kilometer range. Therefore, following the same assumptions for each FSO backhaul

link, the beam width in meters at distance d is obtained as

$$\omega(d) = \omega_0 \sqrt{1 + \left(1 + \frac{2\omega_0^2}{\rho^2(d)}\right)\left(\frac{\lambda_o d}{\pi \omega_0^2}\right)^2}, \tag{2.23}$$

where ω_0 signifies the beam waist radius, and the coherence length $\rho(d)$ is given by $\rho(d) = \left(0.55 C_n^2 k^2 d\right)^{-3/5}$, dependent on $k = \frac{2\pi}{\lambda_o}$ with λ_o as the optical wavelength, and $C_n^2 \approx C_0^2 \exp(-\frac{h}{100})$ is the index of refraction structure parameter at DBS height h, while $C_0^2 = 1.7 \times 10^{-14}$ represents the ground nominal value of refractive index. To proceed further, the author highlights that in this work, it is assumed that the beam is perpendicular to the receiver's lens plane. This assumption is valid for a majority of conventional FSO transceiver architectures mounted on UAVs. These often utilize mechanisms such as gimbals, mirrors, and rotating heads, or a combination thereof, to ensure alignment [44]. Then, given a lens radius r_0, the coefficient $v_1 = \frac{r_0}{\omega_L}\sqrt{\frac{\pi}{2}}$ is used to determine the maximum fraction of optical power captured by the receiver lens, denoted by

$$A_0 = (\mathrm{erf}(v_1))^2 . \tag{2.24}$$

Finally, the GML is given by

$$h_g \approx A_0 \exp\left(-\frac{2u_g^2}{\zeta_{\mathrm{FSO}} \omega_d^2} y\right), \tag{2.25}$$

where u_g represents the misalignment distance (in meters) between the centers of the beam footprint and the receiver's lens. The parameter ζ_{FSO} is selected from the interval $[\zeta_1, \zeta_2]$, providing bounds on the GML loss. In this work, ζ_{FSO} is chosen as the arithmetic mean, i.e., $\zeta_{\mathrm{FSO}} = \frac{\zeta_1 + \zeta_2}{2}$. In [43], the authors prioritize the statistical variations of u as the primary source of GML loss fluctuations. They model the position and orientation perturbations of the DBS - resulting from atmospheric air fluctuations and DBS vibrations - as independent distributed Gaussian RVs. Consequently, definitions are given as $\lambda_1 = \sigma_y^2 + d^2\sigma_\theta^2$ and $\lambda_2 = \sigma_z^2 + d^2\sigma_\phi^2$, where σ_y and σ_z denote the variances of location fluctuations perpendicular to the beam and along the height axis, respectively, while σ_θ and σ_ϕ indicate variances in orientation fluctuations due to rotations around vertical and horizontal axes, respectively. Then, the rate of an FSO link is given by

$$R_{\mathrm{C}} = \frac{1}{2}\log_2\left(\frac{e}{2\pi}\eta_{\mathrm{FSO}}^2 h_p^2 \bar{\gamma} A_0^2\right) - \frac{2}{\zeta_{\mathrm{FSO}} \omega_d^2 \ln(2)}(\lambda_1 + \lambda_2), \tag{2.26}$$

with $\bar{\gamma} = \frac{P_{\text{FSO}}^2}{\sigma_F^2}$ representing the transmit SNR, defined in terms of the FSO transmission power, P_{FSO}, and the noise at the receiver, σ_F. Finally, by constraining the maximum FSO link distance to $d_{\max}$, the achievable throughput of an FSO link between BSs i and l is given by the following equation, where the indices are returned:

$$R_{\text{FSO}}(i,l) = \begin{cases} R_{\text{C}}(i,l) & \text{if } d(i,l) < d_{\max}, \\ 0 & \text{otherwise.} \end{cases} \tag{2.27}$$

2.4 RIS Channel Modeling

RISs have emerged as a transformative technology that holds the potential to enhance wireless communication networks significantly. An RIS consists of electronically controllable surfaces with the ability to alter electromagnetic waves that arrive on them. This capability allows RIS to modify the phase, amplitude, and direction of reflected signals, thereby optimally reshaping the propagation environment to improve signal reception.

One of the primary advantages of RIS is its ability to create favorable propagation conditions, which can lead to more efficient use of the spectrum, enhanced signal quality, and increased coverage areas. This is particularly beneficial in urban environments, where buildings and other structures often cause signal degradation through reflection, diffraction, and scattering. In this thesis in particular, they are leveraged for their capability of creating a virtual LOS path between two communicating nodes that are both visible to the RIS.

Moreover, RIS can help reduce the energy consumption of wireless networks. Unlike traditional relay systems that require power for signal processing and retransmission, RIS passively manipulates the electromagnetic waves without requiring active transmission components. This characteristic not only leads to a reduction in power usage but also decreases the network's operational costs and environmental impact.

The deployment of RIS can also augment the security of wireless networks. By precisely controlling the signal's propagation, RIS can limit the signal's exposure to unintended receivers, thus enhancing the physical layer security. This makes eavesdropping and signal interception considerably more difficult.

Overall, the integration of RIS into wireless networks is expected to play a crucial role in developing next-generation communication systems, including 5G and beyond. Their ability to dynamically control the radio environment promises to overcome traditional challenges in wireless communications, paving the way for more reliable, efficient, and secure wireless connectivity. In this thesis, they are considered for deployment on building facades in urban environments to support backhauling of cells (including DBSs).

2.4.1 Channel model for RIS providing backhaul to BS

The PL in dB between two BSs i and l utilizing a RIS, as per the model in [45], is given by

$$L_{\mathrm{RIS,B}}^{[\mathrm{dB}]}(i,l) = \mathrm{FSPL}_{\mathrm{d}_0} + 10\alpha_{\mathrm{RIS}} \log\left(X_{\mathrm{RIS}}^2(d_1+d_2)\right) - g_{bf}, \tag{2.28}$$

where X_{RIS} represents the count of meta-surfaces within the RIS, and d_1 and d_2 are the distances from the RIS to BS i and l respectively. The model includes the beamforming gain of the RIS, denoted by g_{bf}. In this work, $X_{\mathrm{RIS}} = 3$, and $\alpha_{\mathrm{RIS}} = \alpha_{\mathrm{mm}} = 2.13$.

2.5 Energy Modeling

Energy management is crucial for the effective deployment and operation of DBSs. The limited battery life of these aerial platforms restricts their operating time and affects the continuity of service they provide. Consequently, designing energy-efficient algorithms is essential to maximize the operational duration and optimize coverage, ensuring reliable network service. Typically, a UAV's energy lasts less than 30 minutes, with propulsion accounting for over 95% of a DBS's power usage [46].

Accurate modeling of energy consumption is fundamental in developing effective DBS algorithms. It shapes the planning of flight paths and service strategies and is vital for assessing DBS performance in realistic conditions. Given that battery capacity is a primary limiting factor, integrating precise energy models into DBS simulations is crucial to fully understand their capabilities and limitations in providing persistent aerial network support.

Furthermore, incorporating energy harvesting technologies, such as solar panels, can significantly enhance the operational capabilities of DBSs. This approach not only supports sustainable operations but may also enable extended or even continuous flight operations during suitable conditions. Simulation of energy harvesting effectiveness is as crucial as that of consumption, as it profoundly impacts the design and viability of long-term DBS deployment strategies.

Below, the author details the models used to simulate both the aerodynamic energy consumption of drones and the energy harvested through PV panels.

2.5.1 Energy consumption

According to [47], the energy consumption of a DBS maintaining a constant altitude is described by the equation:

$$P_{\mathrm{DBS}} = P_{\mathrm{level}} + P_{\mathrm{drag}}, \tag{2.29}$$

where P_{level} represents the power required for level flight and is defined by the formula:

$$P_{\text{level}} = \frac{\Omega_W^2}{\sqrt{2}\rho_a A_D \sqrt{||(v_x, v_y)||^2 + \sqrt{||(v_x, v_y)||^4 + \frac{\Omega^2}{(\rho_a A_D)^2}}}}, \tag{2.30}$$

Here, Ω_W denotes the DBS's weight, A_D the total rotor disk area, ρ_a the air density, and (v_x, v_y) the horizontal velocity components. The second term, P_{drag}, accounts for the power consumed due to aerodynamic drag and is calculated as:

$$P_{\text{drag}} = \frac{C_{\text{D0}}\rho_a A_D ||(v_x, v_y)||^3}{8}, \tag{2.31}$$

where C_{D0} is the profile drag coefficient. It is important to note that while P_{level} decreases with increasing horizontal speed, P_{drag} increases, as demonstrated in (2.30).

2.5.2 Energy harvesting

To assess the gain in using PV panels on drones' energy, the power output of a single panel, denoted as P_{PV}, is calculated assuming the panel's inclination is 0°, following the model presented in [48]:

$$P_{\text{PV}} = T_{\text{atm}} \cdot I_0 \cdot \delta_{\text{cloud}} \cdot \eta_{\text{PV}} \cdot S_{\text{PV}} \cdot \sin(\alpha_{\text{solar}}), \tag{2.32}$$

In this formula, α_{solar} represents the solar altitude angle in degrees. The efficiency factor and surface area of the PV panel are given by η_{PV} and S_{PV}, respectively. The term $T_{\text{atm}} \cdot I_0$ describes the incident solar irradiation per square meter outside the atmosphere, measured in W/m^2, where I_0 is the average solar irradiation and T_{atm} accounts for atmospheric transmittance. Additionally, δ_{cloud} is a random factor that adjusts for the influence of cloud cover on solar energy harvesting, which is taken as a random value between 0.8 and 1, reflecting the variability in cloud coverage.

Chapter 3

Fronthaul-Aware DBS Placement

The opening chapter of this thesis explored how DBSs provide on-demand, cost-effective coverage in regions devoid of terrestrial infrastructure or in scenarios demanding enhanced capacity, such as areas experiencing flash mobs or high user density. Section 1.4 outlined the challenges and considerations essential for designing wireless networks that utilize DBSs as cells to provide coverage. This chapter deals with the placement of DBSs to optimize fronthaul-related metrics while discarding the backhaul aspects of the problem. Two solutions are presented: the first, leveraging a modified version of the well-known EM clustering algorithm [49] to place DBSs given the GNs locations. On the other hand, the second algorithm, which uses a multi-agent Q-learning scheme [50], locates DBSs without any knowledge of GNs locations, but based on their SINR measurements. Finally, the chapter introduces a mathematical model for estimating the probability that vehicular users in an urban street lane receive sufficient resources and power when served by a beamformed DBS. This model enables mapping between system parameters, placement strategies, and the resulting quality of service (QoS).

Some of the contributions in this chapter were published in:

1. Janji, S. & Kliks, A. *Multi-Agent Q-Learning for Drone Base Stations*. In: Proc. 2023 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 261–266, 2023.

2. Janji, S. & Kliks, A. *Drone Base Stations Transmission Power Control and Localization*. In: Mobile and Ubiquitous Systems: Computing, Networking and Services, LNICS, vol. 492, Springer, pp. 356–377, 2023. https://link.springer.com/10.1007/978-3-031-34776-4_19

3. Janji, S. & Kliks, A. *Energy-Efficient User Clustering for UAV-Enabled Wireless Networks Using EM Algorithm*. In: Proc. 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), pp. 1–6, 2021.

3.1 Related work

The problem of deciding the locations of DBSs has been extensively studied in the past years, as summarized in the following surveys [5, 1, 10, 11, 51]. Some notable studies are detailed below.

Single DBS Deployment

In terms of single DBS deployment, several researchers have addressed this topic. The work in [52] solved the problem by formulating it as a circle placement and smallest enclosing circle problem, such that the transmit power is minimized while maximizing coverage. A similar approach was adopted in [53], but while maximizing the coverage of users with different SNR. The authors in [38] present the channel model that was outlined in 2.2.1, and use it to obtain the optimal height for a DBS to achieve maximum coverage. In the case where a DBS is supplemented with PV panels, the authors in [47] designed the optimal trajectory for the DBS which first climbs up to a high altitude to harvest a sufficient amount of solar energy and then descends again to a lower altitude to reduce the path loss of the communication links to the GNs it serves, thereby maximizing the system's sum throughput. A different approach is adopted in [54] where the authors formulate the drone-cell placement problem as a Quadratically-Constrained Mixed Integer Non-Linear Optimization problem. This formulation considers the complex relationship between a DBS position and its effective coverage area, and it is solved efficiently using a bisection search method.

Multiple DBSs Deployment

Deploying multiple DBSs introduces additional complexity, such as interference between different clusters of GNs. Numerous solutions have been proposed, such as deploying DBSs along a spiral path to minimize their number while effectively covering users, as suggested in [55]. In [56], the authors consider a graph model in which the edges represent the streets and only outdoor users are considered by representing their densities along the vertices of the graph. An unconstrained optimization problem is formulated whose objective is a combination of three functions: the first calculates the total coverage achieved by DBSs. At the same time, the other two are heuristic functions that introduce a penalty based on the

interference between DBSs and the distance to the nearest charging pole. The problem is solved distributively so that each DBS moves only when its local objective function can be improved.

Machine learning and especially reinforcement learning (RL) has been widely applied to tackle the multiple-DBS deployment problem [5, 51]. In [57], Q-learning and deep Q-learning (DQL) algorithms were used for deploying single and multiple DBSs, respectively, aiming to maximize fairness in coverage while minimizing penalties for collision and interference. Fairness is calculated using Jain's fairness index [58]. The authors of [59] defined state representations, actions, and rewards for training a deep Q-network using prioritized replay, focusing on maximizing user coverage with an obstacle-aware channel model and ignoring interference between DBSs. This is done by first defining a coverage bitmap for obtaining the rewards and states of the proposed RL algorithm. Specifically, the area is divided into a grid, and the number of covered users (i.e., within the coverage range of one or more DBSs) is calculated for each region. Without considering fairness or available energy, the RL algorithm maximizes the number of covered users. Liu et al. in [60] employed deep deterministic policy gradient algorithm (DDPG) to train actor-critic networks for optimizing coverage, fairness, and energy consumption in a continuous space. Unlike other models that consider realistic user distributions, their model assumed coverage needs for predefined points of interest, which might not contain users in practical deployments. A subset of UAVs serve as gateways through satellite links, and a penalty is applied whenever a DBS is far from any gateway and is therefore offline. Also, the work does not consider interference between DBSs. Also, unlike the model used in this thesis, they use a linear energy consumption model, which entails higher DBS power consumption when flying to another location than when hovering at the same spot. This approach is different than more realistic models like the one described in [47], which consider energy savings during hovering flight, and which is adopted in this thesis.

3.2 GNs' Location-Aware DBSs Placement

This section introduces the novel clustering approach for determining the locations of a set of M DBSs with preset transmission power settings, i.e., $P_T(1), \ldots, P_T(M)$. The approach is based on the EM clustering algorithm [49], which is employed to determine the two-dimensional (2D) xy coordinates of each DBS, assuming a uniform fixed height for all DBSs. Further elaboration on the altitude assumptions and their optimization is made in Section 3.2.6. The mechanism is termed SINR-based EM clustering approach (SINR-EM) since it uses an SINR-based metric for updating clusters' locations within the EM framework.

3.2.1 Problem formulation

Given the locations of GNs and the set of DBSs with predetermined transmission power settings described in (2.7), the objective is to find DBSs locations $\{\boldsymbol{m}_j \forall j \in 1,\ldots,M\}$ that maximize the system's EE given by (2.16). The problem is illustrated in Figure 3.1. The goal is to optimize the clustering of GNs with the objective

$$\max_{\boldsymbol{m}_j} \quad \zeta. \tag{3.1a}$$

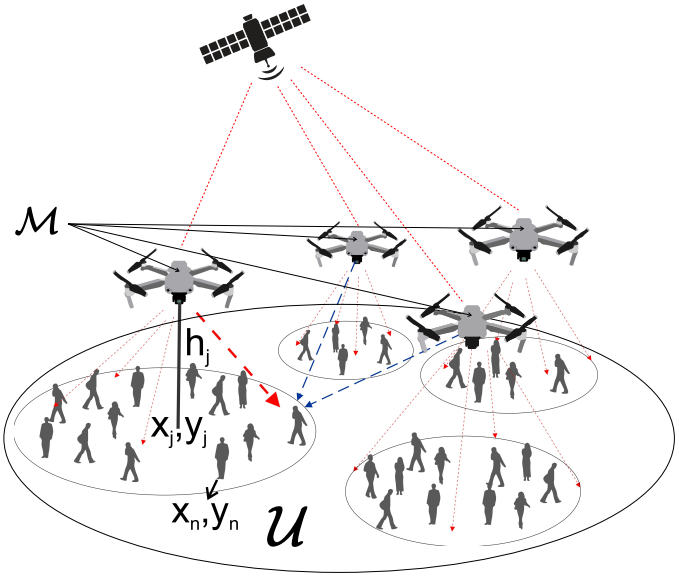


Figure 3.1 Given the locations of GNs, the goal is determine the locations of DBSs and, as a result, the clustering of GNs into cells served by different DBSs. Interference between cells in the downlink direction is considered by the SINR metric. Backhaul is neglected and can be assumed to be available through satellite links.

Currently, the SINR threshold Γ_{th} is discarded in this formulation, focusing instead on the relationship between predefined transmission power settings and the placement of DBSs, given the locations of GNs. A mechanism for controlling transmission power, which incorporates Γ_{th} as a constraint, is discussed later in the next chapter. This approach, combined with the proposed SINR-EM, optimizes service to GNs in terms of Γ_{th} .

3.2.2 NP-hardness

The primary influence on the objective function in (3.1) stems from the distances between GNs and their assigned DBSs. A simpler version of the problem can be considered where all

DBSs share the same transmission power settings and neglect the interference effects. Then, maximizing EE reduces to finding a clustering that minimizes the distances between GNs and their serving DBSs. Assuming a grid of points in 2D space as potential locations from which M DBSs are to be selected, this setup mirrors the p-median problem [61].

The p-median problem can be formally described as follows. Given:

- Demand points $\mathcal{U} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_U\}$.
- Potential facility locations $F = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_N\}$.
- Distance matrix $C = [c_{ij}]$ where c_{ij} is the distance between $\boldsymbol{u}_i$ and $\boldsymbol{f}_j$.
- Integer M representing the number of facilities to locate.

A mapping binary decision variable, $\nu_j \in \mathbb{B}$, is defined, which equals 1 if facility f_j is selected, 0 otherwise. The objective is to minimize the total distance between demand points and the nearest selected facility. The nearest facility is indicated by ξ_{ij}, which is set to 1 if facility j is selected and is closest to demand point i. That is,

$$\min_{\nu_j} \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \xi_{ij} \tag{3.2a}$$

$$\text{s.t.} \sum_{j=1}^{m} \nu_j = M \tag{3.2b}$$

The objective aims to minimize the aggregate distances by optimally locating supply centers for predefined demand points, similar to the reduced version of the targeted problem. The only difference comes from the nonlinearity of the PL function outlined in (2.6) with respect to the distance, which further complicates the problem. However, approximating the problem with a linear heuristic metric similar to (3.2a) is still a reasonable approach. The p-median problem is recognized as NP-hard [62], indicating the impracticality of exact solutions due to potential exponential complexity. Consequently, rather than seeking an exact solution, the author employs the heuristic clustering SINR-EM approach. This heuristic clustering approach considers interference between DBSs and accounts for different transmission power settings by modulating the proposed similarity metric accordingly.

Alternatively, a set of DBSs can be modeled as a set of 2D disks, where each disk has a radius equal to the coverage range of a given DBS. The range is defined as the maximum GN-to-DBS distance beyond which the GN receives insufficient signal power (i.e. below receiver sensetivity). The problem of determining the number and placement of such coverage disks

to ensure that all GNs are covered can thus be formulated as a geometric set cover problem (GSC) problem [63]. This class of problems is also known to be NP-hard.

3.2.3 Expectation-Maximization (EM)

EM algorithm often employs Gaussian mixture models (GMMs), modeling a set of sample points as originating from a *mixture* of Gaussian distributions [49]. A GMM determines the likelihood of a sample point originating from this mixture, defined as

$$\pi(\boldsymbol{u};\theta) = \sum_{j=1}^{M} P_j^{\mathrm{EM}} \mathcal{N}(\boldsymbol{u};\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j), \tag{3.3}$$

where the parameters $\theta = \{P_1^{\mathrm{EM}},\ldots,P_M^{\mathrm{EM}},\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_M,\boldsymbol{\Sigma}_1,\ldots,\boldsymbol{\Sigma}_M\}$ include mixing proportions $\boldsymbol{P}^{\mathrm{EM}}$, means $\boldsymbol{\mu} \in \mathbb{R}^2$, and covariance matrices $\boldsymbol{\Sigma} \in \mathbb{R}^{2\times 2}$ of Gaussian components. $\mathcal{N}(\boldsymbol{u}\,;\boldsymbol{\mu}_j,\boldsymbol{\Sigma}_j)$ represents the probability that the sample point $\boldsymbol{u}$ is taken from the j-th Gaussian distribution and is given by (2.3).

EM learns these parameters θ from a set of sample points, $\mathcal{U}$, and the number of mixtures, M. The notation for the number of Gaussian components, M, and the set of sample points, $\mathcal{U}$, is consistent with the mapping of these parameters to the total number of DBSs and the 2D locations of GNs, respectively. Moreover, each mean $\boldsymbol{\mu}_j$ corresponds to a DBS j's 2D location $\boldsymbol{m}_j$. The algorithm iteratively performs two steps

1. Estimation step (E-step). Evaluate the posterior assignment probabilities as:

$$p^{(l)}(j|n) = p(j|\boldsymbol{u}_n,\theta^{(l)}) = \frac{P_j^{\mathrm{EM},(l)} N\left(\boldsymbol{u}_n;\boldsymbol{\mu}_j^{(l)},\boldsymbol{\Sigma}_j^{(l)}\right)}{\pi(\boldsymbol{u}_n;\theta^{(l)})}, \tag{3.4}$$

   where (l) denotes iteration number, and $\boldsymbol{u}_n = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$ is the n-th sample, with U as the total number of samples.

2. Maximization step (M-step). Update parameters to enhance the model's likelihood according to:

$$P_j^{\mathrm{EM},(l+1)} = \frac{\hat{n}(j)}{U}, \text{ where } \hat{n}(j) = \sum_{n=1}^{U} p^{(l)}(j|t). \tag{3.5a}$$

$$\boldsymbol{\mu}_j^{(l+1)} = \frac{1}{\hat{n}(j)} \sum_{n=1}^{U} p^{(l)}(j|n)\boldsymbol{u}_n. \tag{3.5b}$$

$$\boldsymbol{\Sigma}_j^{(l+1)} = \frac{1}{\hat{n}} \sum_{n=1}^{U} p^{(l)}(j|n) \left(\boldsymbol{u}_n - \boldsymbol{\mu}_j^{(l+1)}\right) \left(\boldsymbol{u}_n - \boldsymbol{\mu}_j^{(l+1)}\right)^T. \tag{3.5c}$$

The E-step calculates the posterior probabilities $p^{(l)}(j|n)$, indicating the likelihood of observing $\boldsymbol{u}_n$ from the j-th Gaussian component. The M-step updates each component j's parameters to maximize the overall likelihood of the GMM. Notably, the EM algorithm is assured to converge to a local optimum [49].

3.2.4 SINR-based Expectation-Maximization algorithm (SINR-EM)

In clustering, similarity metrics are critical in determining how data points are grouped together. These metrics assess the closeness or similarity between individual data points or between a data point and a cluster centroid. Standard metrics include Euclidean distance, Manhattan distance, cosine similarity, and others, each suitable for specific types of data and clustering objectives. In the context of the EM approach described above, the similarity metric fundamentally differs from traditional distance measures. Instead of calculating distances, the EM algorithm uses the PDF of Gaussians as a similarity measure. That is, the probability of a data point $\boldsymbol{u}_n$ indicates the *similarity* of that point to a Gaussian component with given parameters. This probabilistic approach not only assigns data points to the most likely cluster but also handles uncertainties and mixed memberships effectively, where data points can belong to multiple clusters to varying degrees. This is particularly useful in complex data environments where clusters are not distinctly separable and may overlap.

Can we define a similarity measure between a GN location and a specific DBS, given its transmission power and location, to determine the likelihood of association based on the received SINR under a max-SINR association scheme? While Euclidean distance alone is appropriate when all DBSs have identical transmission powers, the intricate relationship between distance and power, as described in (2.8), means that distance-based metrics inadequately represent SINR dynamics in networks where DBSs have varied power settings. For instance, a GN might receive higher power from a DBS transmitting at 200 mW and located

100 m away than from another DBS at 50 m but transmitting at 100 mW. How then can we develop a similarity metric that effectively incorporates the distance/power relationship detailed in (2.8)?

Proposed SINR-based metric

At each iteration (l), given the current cluster centers $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M$ representing DBSs locations, the posterior probabilities are replaced in (3.4) with probabilities directly linked to the SINR values specified in (2.10). These are calculated using the predefined transmission power settings. The new probabilities for each GN are computed by

$$\boldsymbol{q'}^{(l)}(\cdot|n) = s\left(\boldsymbol{\Gamma}^{(l)}(n)\right), \tag{3.6}$$

where each element of this vector corresponds to a different DBS, and $\boldsymbol{\Gamma}^{(l)}(n) \in \mathbb{R}^M$ is the vector of SINR values for GN n across all DBSs:

$$\boldsymbol{\Gamma}^{(l)}(n) = \left[\Gamma(1,n)^{(l)}, \Gamma(2,n)^{(l)}, \ldots, \Gamma(M,n)^{(l)}\right]^T, \tag{3.7}$$

and $s(z)$ is the vector resulting from the Softmax function with entries defined as

$$s(z)_j = \frac{e^{\varkappa z_j}}{\sum_{j'=1}^{M} e^{\varkappa z_{j'}}}, \tag{3.8}$$

with $\varkappa$ being a scaling constant. This adaptation in (3.6) allows user contributions to each DBS's parameter update to be weighted by the relative likelihood of service from that drone compared to others, directly proportional to the SINR achieved as detailed in (2.10).

In addition, a dynamic SINR threshold $\hat{\Gamma}_{\text{th}}^{(l)}(n)$ is introduced below which $q(j|n)$ will be zeroed if $\Gamma(j,n) < \hat{\Gamma}_{\text{th}}^{(l)}(n)$, defined as

$$\hat{\Gamma}_{\text{th}}^{(l)}(n) = \max\left\{\boldsymbol{\Gamma}^{(l)}(n)\right\} - 10^{\frac{S_{\text{th}}}{10}}, \tag{3.9}$$

where the highest achieved SINR among all drones is reduced by S_{th} in linear scale, the SINR dropout value. This determines the rate at which users disconnect from non-serving drones and do not contribute to their location updates. The decoupling mask is then

$$\boldsymbol{M}^{(l)}(n) = \left[H\left(\Gamma(1,n)^{(l)} - \hat{\Gamma}_{\text{th}}^{(l)}(n)\right), \ldots, H\left(\Gamma(M,n)^{(l)} - \hat{\Gamma}_{\text{th}}^{(l)}(n)\right)\right], \tag{3.10}$$

and the normalized masked similarity values (or probabilities) are calculated as

$$\boldsymbol{q}^{(l)}(\cdot|n) = \frac{\boldsymbol{M}^{(l)T}(n) \odot \boldsymbol{q'}^{(l)}(\cdot|n)}{\|\boldsymbol{M}^{(l)T}(n) \odot \boldsymbol{q'}^{(l)}(\cdot|n)\|}, \tag{3.11}$$

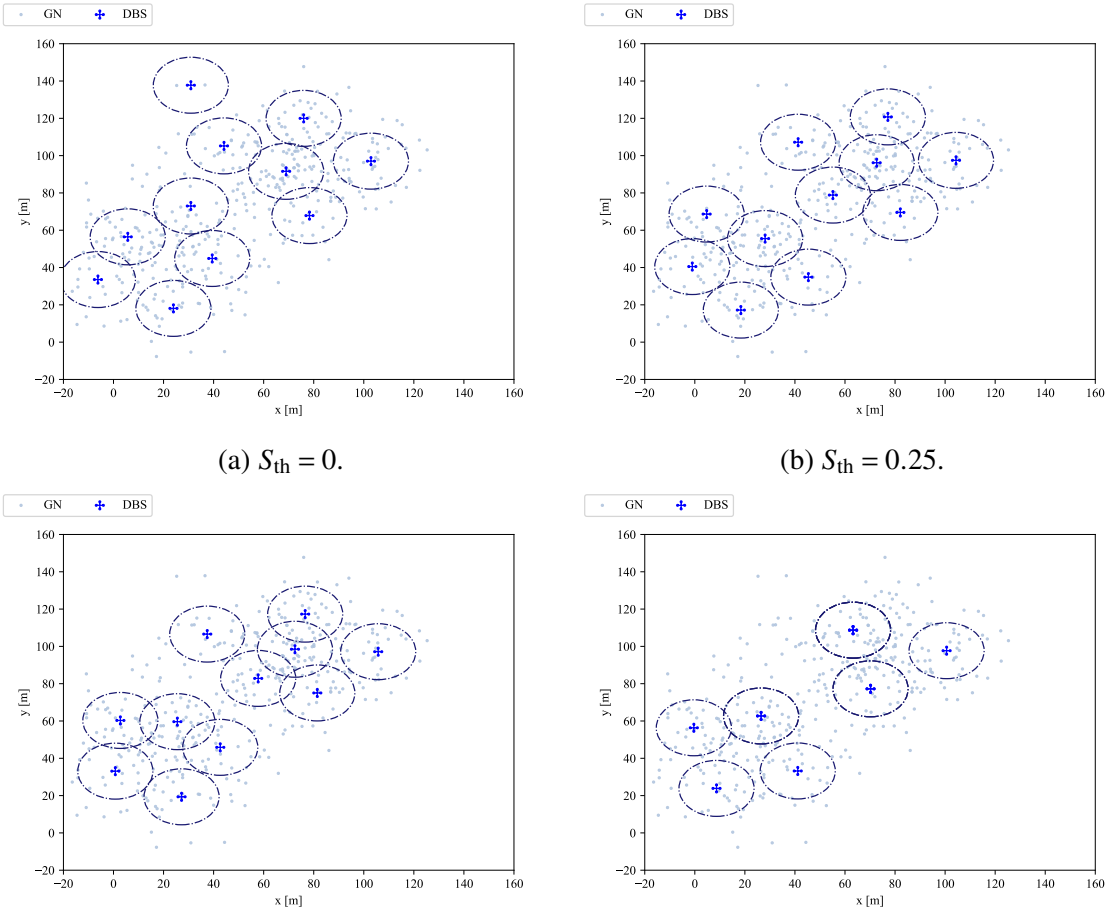
where $\odot$ indicates element-wise multiplication, and $\|\cdot\|$ is the L2 norm, ensuring the probabilities sum to one after discarding DBSs below the threshold.

Given this novel similarity, the SINR-EM approach then consists of obtaining probability-like values for all GNs in the estimation step by using (3.11) instead of (3.4), and then updating the cluster center by applying (3.5b) in the maximization step. That is, there is no longer a need to obtain the mixing proportions and covariance estimates.

The pseudocode for SINR-EM algorithm is shown in Algorithm 1 where the clusters' locations are initialized using a set of unique initial locations μ_{init}. Furthermore, the convergence is determined whenever an iteration results in a mean update less than some predefined threshold ϵ for all clusters.

Algorithm 1 SINR-Based EM Algorithm.

Input: $\mathcal{U}$, M, $\boldsymbol{P}_T$
Output: $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, .., \boldsymbol{\mu}_M]$
1: *initialisation*: $\boldsymbol{\mu} = \boldsymbol{\mu}_{\text{init}}$
2: **repeat**
3: Calculate SINR-based similarity values using (3.11)
4: Obtain new DBSs 2D locations using (3.5b)
5: **until** convergence



(a) $S_{\text{th}} = 0$.

(b) $S_{\text{th}} = 0.25$.

(c) $S_{\text{th}} = 0.5$.

(d) $S_{\text{th}} = \infty$.

Figure 3.2 Comparison of SINR-EM resulting clusters overlap for different values of S_{th}. Notice how setting $S_{\text{th}} = 0$ results in one cluster being located on top of two outlier GNs, while increasing S_{th} to 0.5 further attracts DBSs to hotspots, thus increasing their interference. Eventually, for large values of S_{th} we observe that some clusters are merged and therefore a collision between multiple DBSs occurs.

3.2.5 Choosing the dynamic SINR threshold

S_{th} regulates the degree of membership blending for GNs across various clusters, influencing the aggregation or separation of cluster members. By setting $S_{\text{th}} > \epsilon$, where $\epsilon > 0$, the range of points that influence the centroid updates of cluster j is broadened, thus incorporating more members. This approach is particularly useful when dealing with outliers in the dataset. Without this adjustment, a cluster centered around outliers might become isolated, continuously updating its location based solely on these outliers and not integrating more central members of the data distribution. To mitigate this, increasing S_{th} allows the cluster

to include members positioned further from the current outlier-centric location, thereby facilitating more representative centroid updates. Conversely, a higher S_{th} increases the likelihood of DBSs being drawn towards areas with higher densities of GNs, potentially leading to closer proximity between clusters. This introduces a trade-off between minimizing the influence of outliers and reducing the overlap (also interference) among clusters due to increased shared memberships.

Increasing S_{th} indefinitely can lead to clusters overlapping extensively, resulting in their centers converging. This scenario increases the risk of collisions between DBSs flying at the same altitude. Collision avoidance is crucial in multi-UAV systems [64], and various algorithms have been developed to manage UAVs' movements and prevent collisions [65]. In the context of SINR-EM, setting a smaller S_{th} ensures more disjoint cluster memberships, thus maintaining a safer distance between different DBSs. Figure 3.2d illustrates these dynamics by displaying the impact of varying S_{th} values on the spatial distribution of GNs in SINR-EM.

3.2.6 Optimizing the Altitude of DBSs

In this thesis, the optimization of DBSs altitude is considered of secondary importance due to various reasons. Regulatory limits often cap the maximum altitudes. For example, the EASA limits altitude to 120 meters [6], the Federal Aviation Administration (FAA) to approximately 122 meters (400 feet) [66], and French regulations restrict it to as low as 50 meters in some areas [67]. Additionally, the energy required for ascending to higher altitudes significantly surpasses that for maintaining a steady horizontal flight at a fixed altitude, which indicates that energy is conserved when altitude is limited [47]. Furthermore, lower altitudes can also reduce interference between DBSs and other network nodes [68]. Also, in certain operational scenarios, maintaining a fixed, lower altitude may simplify the trajectory control framework by minimizing the need to navigate around obstacles [7]. Despite these constraints, the author explores a mechanism to fine-tune the altitudes of DBSs after their 2D positioning, using the approach described in [38]. This can be done to improve the SNR of served users, or to adapt to their mobility by extending and shrinking the coverage disks of DBSs.

The PL model discussed in [38], which is adopted in this thesis for simulating the DBS-GN fronthaul channel as described in Section 2.2.1, can help in finding the optimal height. The authors of [38] have developed an optimization problem aimed at identifying the maximum altitude that maximizes the coverage radius $R_{\text{A}}(j)$ of a DBS given specific environmental parameters and an upper PL limit $L_{\max}$. They showed that for any given set of environmental conditions, a fixed elevation angle $\omega_{\text{opt}} = \arctan\left(\frac{h_j}{R_{\text{A}}(j)}\right)$ accurately describes the relationship between a DBS's altitude and the maximal coverage radius achievable within

the specified $L_{\max}$. It turns out, this angle remains constant across different $L_{\max}$ values for a certain environment. This relationship is illustrated in Figure 3.3.

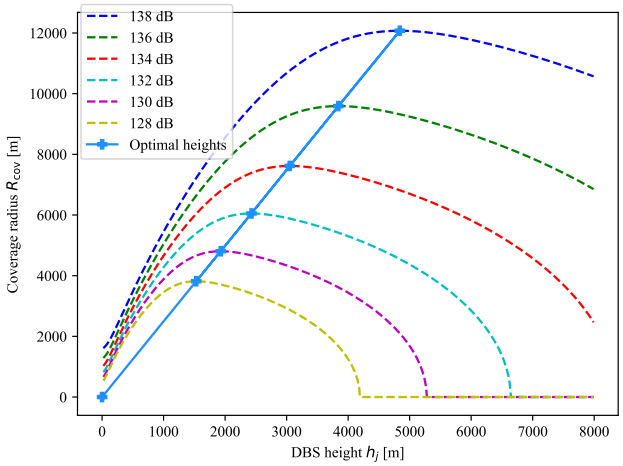


Figure 3.3 The figure shows the coverage radius $R_{\text{A}}(j)$ achieved for given DBS height, assuming a certain $L_{\max}$ threshold. It turns out that the ratio between height and radius is always constant, corresponding to a fixed ω_{opt} depending only on the environment parameters.

To find ω_{opt} we need to solve the following equation [38, Eq. (13)]:

$$\frac{\pi}{9\ln(10)}\tan(\omega_{\text{opt}}) + \frac{\alpha\beta(\eta_{\text{LOS}} - \eta_{\text{NLOS}})\exp(-\beta(\omega_{\text{opt}} - \alpha))}{[\alpha\exp(-\beta(\omega_{\text{opt}} - \alpha)) + 1]^2} = 0. \tag{3.12}$$

After obtaining ω_{opt}, to optimize the height of a DBS j the author first calculates its coverage radius $R_{\text{A}}(j)$ by finding the distance to the farther GN it serves. That is,

$$R_{\text{A}}(j) \geq \max_{\forall n \in \{1,\ldots,U\}} d_{\text{2d}}(j,n)\, W(j,n) \tag{3.13}$$

where the 2D distance function in (2.2) is used along with the max-SINR mapping variable defined in (2.12). Then, the height is set to the value

$$h_j = \max\left(h_{\min}, \min\left(h_{\max}, \tan(\omega_{\text{opt}}) R_{\text{A}}(j)\right)\right) \tag{3.14}$$

where it was limited to the range defined by $[h_{\min}, h_{\max}]$. Thus, due to this relationship between the height and coverage radius, the height can also be varied to extend the coverage disk further when GNs move as described later. However, for maximum permissible path loss $L_{\max}$, and the bounds $[h_{\min}, h_{\max}]$, there exists a corresponding maximum coverage

radius $R_{\max}$, determinable using [38, Eq. 12]. This value is the upper limit of $R_{\mathrm{A}}(j)$. Therefore, if a GN moves beyond $R_{\max}$, the only solution is to reposition the DBSs.

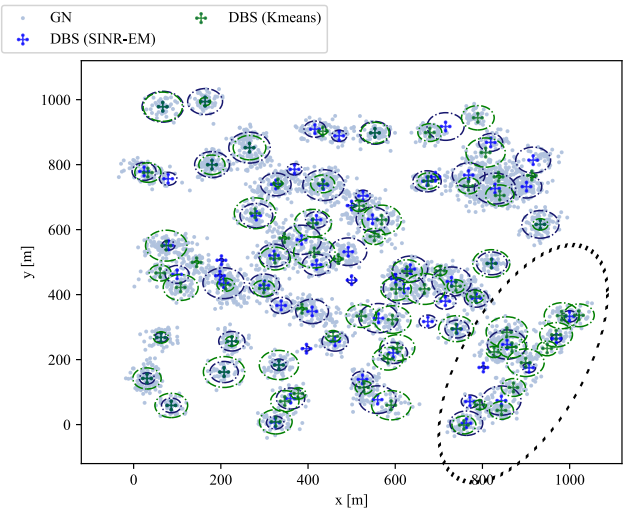


Figure 3.4 An example with TCP-generated GNs. A set of 70 DBSs with random transmission power settings is placed either according to SINR-EM (highlighted in blue), or by using k-means++ (highlighted in green). Each DBS is also encircled with a radius that is proportional to its transmission power. Notice how k-means++ tends to cluster DBSs with higher transmission powers closer together, leading to significant overlaps. Conversely, SINR-EM strategically spaces out DBSs based on their power settings.

3.2.7 Simulation and Discussion

This section demonstrates the effectiveness of the SINR-EM algorithm by comparing it with the well-known k-means clustering algorithm with k-means++ seeding (k-means++) [69]. Unlike k-means++ and traditional clustering methods, SINR-EM integrates the transmission power settings of DBSs into its clustering process. This integration allows for the adjustment of cluster sizes according to the transmission power of the serving DBS. Simulations are conducted within a 1 km^2 area, generating 1000 different scenarios where GNs distributions are modeled using the TCP method outlined in Section 2.1.1. For each simulation run, the transmission power settings for the DBSs, measured in Watts, are randomly selected from the set $\{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95\}$, and the heights of DBSs are set to a uniform value of 25 m. Furthermore, $B_n = 500$ kHz, $B_D = 20$ MHz, $\sigma_0^2 = -147$ dB, and $f_c = 2$ GHz. As for the PL environment variables, an urban environment is assumed with $\alpha = 9.61$, $\beta = 0.16$, $\alpha = 9.61$, $\eta_{\mathrm{LOS}} = 1$, and $\eta_{\mathrm{NLOS}} = 20$.

Figure 3.4 illustrates the placement outcomes for one particular instance, comparing SINR-EM with k-means++. In the figure, each DBS is encircled with a radius that is

proportional to its transmission power. This visualization highlights a critical distinction: whereas k-means++ may position DBSs in close proximity irrespective of their power settings, SINR-EM strategically spaces out DBSs with higher transmission powers, effectively minimizing potential interference and optimizing coverage.

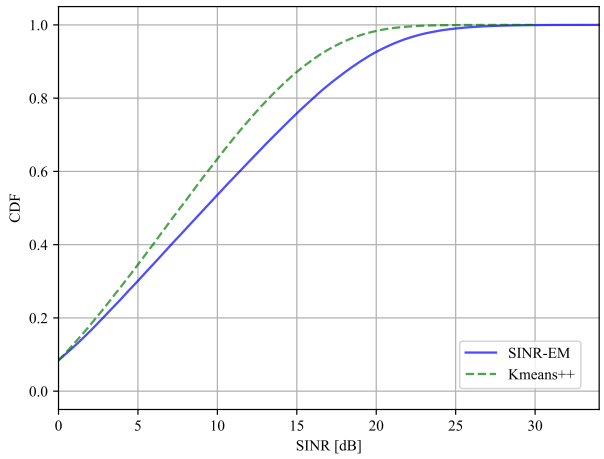
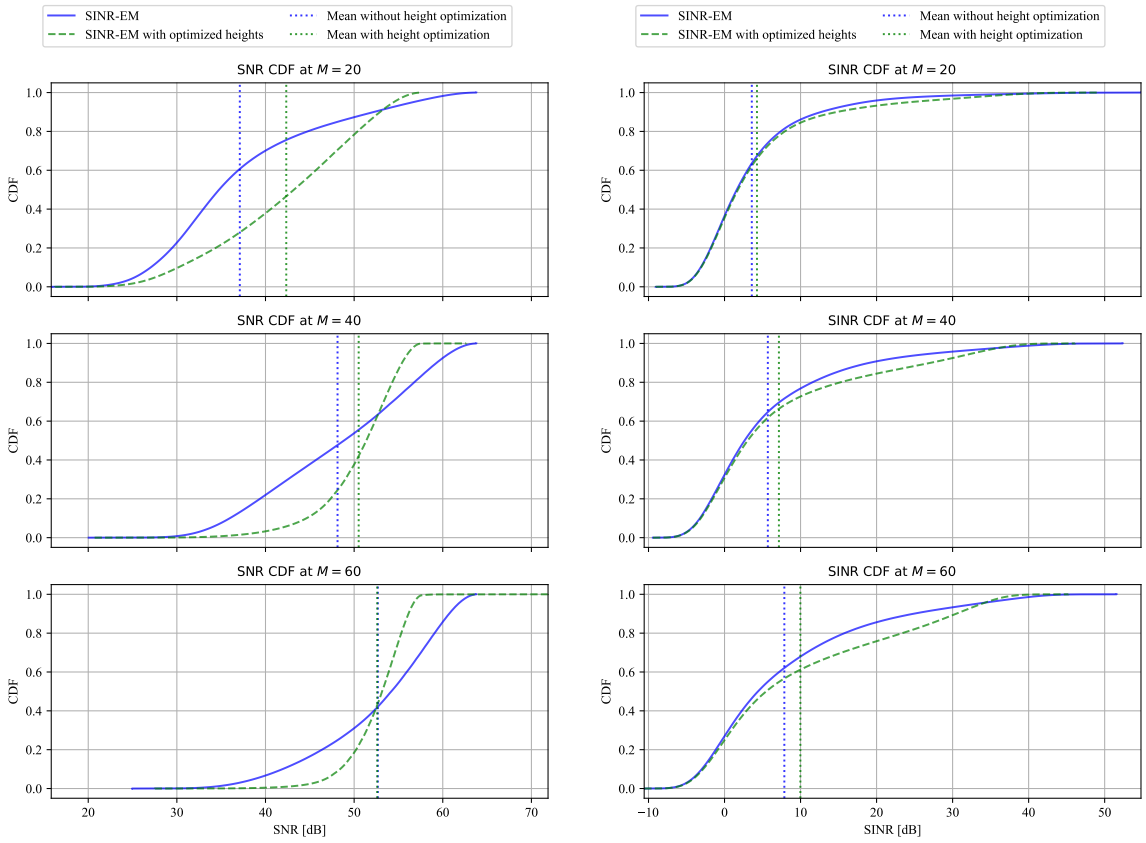


Figure 3.5 A CDF plot of the aggregated SINR values of all GNs across 1000 runs. The simulations are run with 70 DBSs, $S_{\text{th}} = 0.5$, and $\varkappa = 100$. Clearly SINR-EM outperforms k-means++ since it reduces the interference and takes into consideration

Figure 3.5 plots a cumulative distribution function (CDF) of the achieved SINR values for all GNs across all runs. As expected, an improvement in the average link quality is observed when employing SINR-EM as it reduces the interference between clusters.

The final results shown are related to fine-tuning the heights of DBSs after determining their 2D locations using (3.12), (3.13), and (3.14). Since the height optimization mechanism proposed improves the SNR without considering interfernce, the author compares both SNR and SINR results with and without height optimization for $h_{\text{min}} = 10$ m and $h_{\text{max}} = 50$ m, and for different DBSs count (M). The results are plotted in Figure 3.6. As observed, increasing the number of deployed DBSs increases the achieved SNR of users. Also, optimizing the height improves the minimum SNR achieved, and for smaller M, the average SNR of all GNs as well. On the other hand, the SINR performance seems to be enhanced slightly with no substantial variation of results.

Later in the next chapter, more results are provided on the performance of SINR-EM when combined with a power control algorithm, and the performance is compared with another baseline method which is presented in literature.



(a) SNR CDF. Optimizing the height clearly improves the minimum SNR achieved in all cases, while the mean SNR is improved only for small M.

(b) SINR CDF. A slight improvement in overall SINR values is observed.

Figure 3.6 A CDF plot of the aggregated SNR (left) and SINR (right) values of all GNs across 1000 runs for different DBSs count (M). The mean values also shown by horizontal lines. The simulations are run with $S_{\text{th}} = 0.5$, $\varkappa = 100$, $h_{\text{min}} = 10$ m and $h_{\text{max}} = 50$ m.]

3.3 Location-Unaware DBS Placement

In contrast from traditional approaches that require GNs location data, the author developed a multi-agent RL strategy for DBSs placement that does not rely on predefined geographic grids or coverage of imaginary fixed points. Instead, the proposed method leverages a history of channel measurements collected during RL episodes to inform placement decisions. This approach is demonstrated in a simulated urban environment where user movements are realistically modeled using the OMM described in Section 2.1.2.

Each DBS is assigned a set of potential locations from which it selects a position each cycle, either moving to a new location or remaining stationary if the same location is chosen again. The criteria for defining possible locations of DBSs that was presented in Section 3.3.3 ensure that every potential arrangement of locations meets a specified minimum SNR threshold across the entire area. In contrast to other methodologies that primarily focus on SNR [57, 59], the proposed decision-making process incorporates SINR metrics. The proposed approach uses a novel multi-agent RL framework, as proposed in [50], designed to efficiently handle problems with multiple agents.

3.3.1 Problem formulation

Figure 3.7 illustrates the scenario under consideration, featuring M DBSs providing coverage over an urban area populated by U users who move according to the OMM detailed in Section 2.1.2. In this study, the backhaul connections of DBSs are ignored and presumed to be maintained through either satellite or FSO links. Each DBS has k predetermined potential locations at uniform heights, from which it selects one during each decision period (for instance, $k = 2$ as depicted in Figure 3.7). The set of potential locations for DBS j is represented by $\mathcal{K}_j$. Contrary to Chapter 2 where time indices were omitted, here, the current position of DBS j at time t is expressed as $\mathbf{m}_j(t) = (x_j(t), y_j(t), h_j(t))$. Correspondingly, the 2D positions of users are denoted by $\boldsymbol{u}_n(t) = (x_n(t), y_n(t))$, where again, $n \in \{1, 2, .., U\}$. Furthermore, to characterize the DBS-GN channel, the model presented in Section 2.2.1 is used again.

3.3.2 DBS speed and aerodynamic power consumption

Referring back to equation (2.30) from Section 2.5.1, which describes the power consumption for a DBS's level flight, P_{level}, we observe that an increase in the DBS's horizontal flight speed decreases the power required to maintain altitude. This reduction in power for faster speeds arises due to air resistance's effect on the rotor blades, aiding in the DBS's lift as its

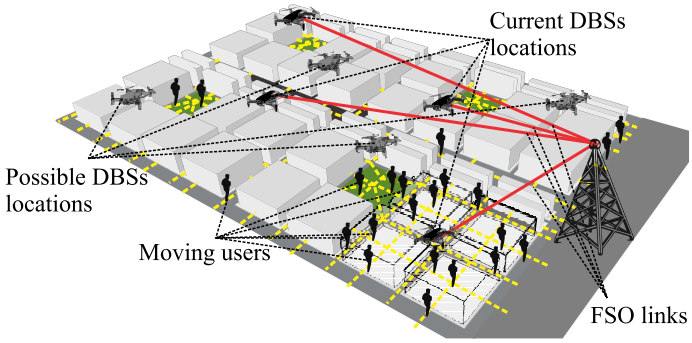


Figure 3.7 Simulation scenario of the proposed RL methodology using an extended Madrid grid configuration. This setup includes four replicated Madrid grids with GNs movements simulated via the OMM outlined in Section 2.1.2. The simulation deploys $M = 4$ DBSs, each with $k_{\mathrm{p}} = 2$ potential locations per cycle. These DBSs are assumed to maintain backhaul connections to a central station (e.g., using FSO).

speed increases (refer to Figure 3.8) due to the power generated from the friction between the air and rotor blade.

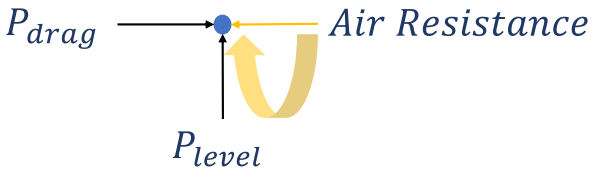


Figure 3.8 Illustration of forces acting on the DBS: P_{level} sustains altitude, while P_{drag} induces horizontal motion. Air resistance, although opposing motion, indirectly reduces P_{level} by enhancing lift through interaction with the drone's rotor blades.

Figure 3.9 demonstrates how increasing the DBS's speed reduces its overall power consumption, even when the speed is up to approximately 25 meters per second. Beyond this speed, the additional power required to overcome drag outweighs the lifting benefits. The least power consumption is achieved at 13 m/s, which is chosen as the preferred speed for DBS mobility. This finding is based on a DBS weight of $\Omega_W = 39.22$N, a rotor disk area of $A_D = 0.78$ m^2, and an air density of $\rho_a = 1.225$ kg/m^3.

This analysis highlights the energy efficiency benefits of moving DBSs. Later, it is shown that strategic movement can also enhance both system fairness and user QoS, as discussed in Section 3.3.9.

3.3.3 Selecting Possible Locations

To ensure complete coverage, the location selection for the M DBSs adheres to the criterion that every point within the operational area must be within the coverage of at least one DBS

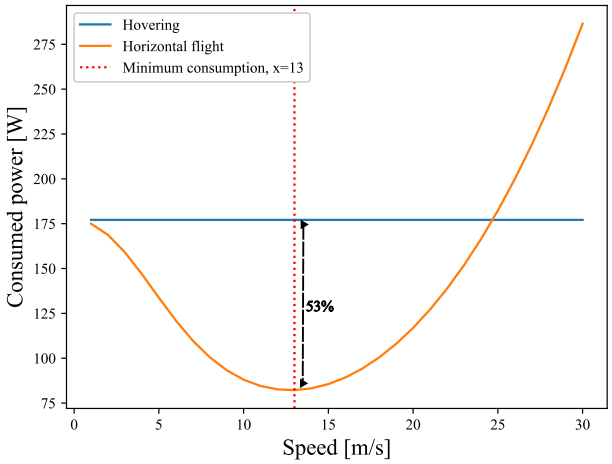


Figure 3.9 The relationship between DBS speed and aerodynamic power consumption. Increased speed enhances lift due to air resistance, significantly reducing power consumption at an optimal speed of 13 m/s, which is 53% more energy efficient than hovering at a fixed location.

meeting the maximum path loss, $L_{\max}$, and during all time instances. The coverage radius, $R_{\mathrm{A}}(j)$, is determined for the DBSs based on the optimal elevation angle, ω_{opt}, derived from (3.12). Then, the area is segmented into M sub-areas, with each designated to be continually covered by a respective DBS. For each DBS j, k potential locations are designated to ensure that all points in its assigned sub-area are within its coverage radius at all times, as depicted in Figure 3.10.

The environmental variables α and β are extracted using the methodology outlined in Section 2.2.1, which considers the height distribution of buildings, the built-up area ratio, and the number of buildings per square kilometer. The obtained values are $\alpha = 5.05$ and $\beta = 0.08$.

3.3.4 Reinforcement Learning and Markov Decision Processes (MDPs)

RL utilizes the Markov decision process (MDP) framework, wherein an agent makes decisions within a stochastic environment to optimize expected rewards. MDPs offer a robust mathematical structure to tackle decision-making challenges where the outcomes are influenced by both random factors and the agent's actions. Specifically, MDPs are useful for modelling tasks that involve discrete-time decision-making under uncertain conditions, provided the agent can fully observe the environmental state [7]. The components of MDP include:

Figure 3.10 Illustration of the division of an area into $M = 4$ sub-areas, ensuring that the DBS locations selected for each sub-area provide continuous coverage as defined by the radius $R_{\text{A}}(j)$.

- **State Space** ($\mathcal{S}_{\text{Q}}$): This represents the environmental conditions affecting decision-making, encompassing all relevant factors such as the location and energy levels of a DBS.
- **Action Space** ($\mathcal{A}_{\text{Q}}$): This is a finite set of all possible actions the agent can take to influence the state, such as choosing the next point to which DBS moves.
- **Transition Function** (P): Defined as $P : \mathcal{S}_{\text{Q}} \times \mathcal{A}_{\text{Q}} \times \mathcal{S}_{\text{Q}} \rightarrow [0, 1]$, it describes the likelihood of transitioning to a new state $\boldsymbol{s}_{t+1}$, given the current state $\boldsymbol{s}_t$ and an action $\boldsymbol{a}_t$.
- **Reward Function** (R): This function maps each action-state pair to a numerical reward, providing a metric for the effectiveness of decisions relative to the agent's goals.

Such a structured approach makes RL especially suitable for sequential decision-making tasks. The RL process cycles through phases where the agent observes the state s_t, executes an action a_t, and receives a reward r_t from the environment, advancing to the next state s_{t+1}. The agent's primary objective is to develop a policy that maximizes cumulative rewards. This cyclic interaction is illustrated in Figure 3.11, highlighting the role of RL in learning and optimizing agent behavior.

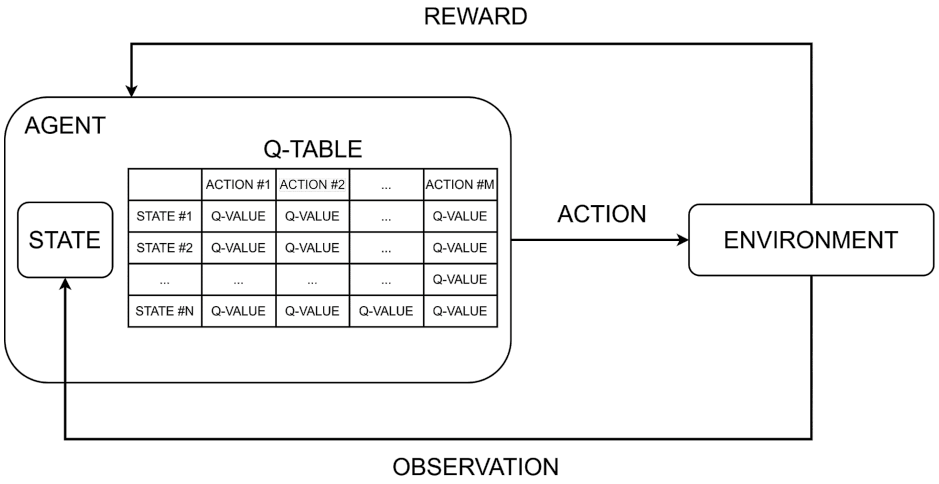


Figure 3.11 Illustration of an agent in an MDP framework interacting with the environment through observing a state, selecting an action, and receiving a reward. The agent's experience is memorized in its Q-table, which is used to select future decisions that maximize the expected rewards.

3.3.5 Q-learning

Q-learning is a well-known off-policy method in RL. This method is described as off-policy because it allows an agent to learn the optimal policy independently of the policy actually being taken by the agent. If all actions are repeatedly sampled, the Q-learning approach is guaranteed to converge to optimal state-action selection criteria [70].

The fundamental concept of Q-learning involves the agent adopting a policy, π, which maximizes expected reward returns. This policy, $\pi(\boldsymbol{a}|\boldsymbol{s})$, represents a probabilistic distribution over actions given the current state, where $\pi : \mathcal{S}_\mathrm{Q} \times \mathcal{A}_\mathrm{Q} \rightarrow [0,1]$. For deterministic scenarios, this reduces to $\pi(\boldsymbol{s})$, mapping each state directly to an action.

The states within a state space are assigned a quality metric through the value function V^π, which is the expected cumulative reward from that state onward under policy π:

$$V^\pi(\boldsymbol{s}) = \mathbb{E}_\pi[G_t|\boldsymbol{s}_t = \boldsymbol{s}] \tag{3.15}$$

where G_t, the return from state $\boldsymbol{s}_t$, is calculated as

$$G_t = \sum_{k=t}^{T} \gamma^{k-t} r(\boldsymbol{s}_k, \boldsymbol{a}_k) \tag{3.16}$$

Here, $\gamma \in [0,1]$ acts as the discount factor, controlling the significance of immediate versus future rewards.

The recursive nature of value functions is shown through the Bellman equation [70]

$$V^{\pi}(\boldsymbol{s}) = \mathbb{E}_{\pi}[r_t + \lambda V^{\pi}(\boldsymbol{s}_{t+1}) | s_t = \boldsymbol{s}] = \sum_{\boldsymbol{s}'} \mathcal{P}(\boldsymbol{s}, \pi(\boldsymbol{s}), \boldsymbol{s}') \left(R(\boldsymbol{s}, \pi(\boldsymbol{s}), \boldsymbol{s}') + \lambda V^{\pi}(\boldsymbol{s}')\right). \quad (3.17)$$

This equation establishes that the value of a state $\boldsymbol{s}$ depends on both immediate rewards and the expected values of future states, adjusted by the state transition probabilities and the discount factor.

For a given MDP, the optimal Q-function, Q^*, which dictates the best course of action from each state, is defined as:

$$Q^*(\boldsymbol{s}, \boldsymbol{a}) = \sum_{\boldsymbol{s}'} \mathcal{P}(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') \left(R(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}') + \lambda \max_{a'} Q^*(\boldsymbol{s}', \boldsymbol{a}')\right). \quad (3.18)$$

With an optimal policy defined as

$$\pi^*(s) = \arg\max_{a} Q^*(\boldsymbol{s}, \boldsymbol{a}). \quad (3.19)$$

Therefore, using the Q-function eliminates the need for forward planning per decision, relying instead on maximizing the Q-values, which already incorporate all necessary information for the decision-making.

During environmental interaction, the agent continually updates its Q-values using the following rule:

$$Q^{\text{new}}(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \left[r_t + \lambda \max_{a} Q(s_{t+1}, a) - Q(\boldsymbol{s}_t, \boldsymbol{a}_t)\right], \quad (3.20)$$

where γ represents the learning rate. Typically, an ε-greedy approach is employed where the agent either exploits the best-known action with probability $1 - \varepsilon$ or explores a new action randomly with probability ε.

Q-learning's popularity in wireless network management is due to its robustness and adaptability to dynamic environments, qualities essential for such applications. Unlike traditional optimization that relies on continually updated system data such as user locations and channel conditions, Q-learning operates effectively with incomplete models. This model-free approach supports the development of adaptive strategies through ongoing interaction with the environment, avoiding the scalability issues associated with conventional methods that often see computational demands spike with increasing input sizes. By defining states appropriately, as demonstrated in the approach, Q-learning ensures scalability irrespective of the complexity of the network. After the learning phase, agents leverage the acquired

model to make real-time decisions. Additionally, the adaptibility of Q-learning allows for the application of learned models across different but similar environments through transfer learning [71].

Double Q-learning

To address the maximization bias arising from the maximization operator in (3.20), the author implemented the double Q-learning method as suggested in [70]. This approach uses two Q tables, Q_1 and Q_2, each holding values for all possible state-action pairs. During decision-making at state $\boldsymbol{s}_t$, the action is selected that maximizes the sum $Q = Q_1(\boldsymbol{s}_t, \boldsymbol{a}_t) + Q_2(\boldsymbol{s}_t, \boldsymbol{a}_t)$. In the training phase, each epoch randomly updates one of the tables with equal probability. The update process is as follows.

$$Q_1(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q_1(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \left[r_t + \lambda Q_2 \left(s_{t+1}, \arg\max_a Q_1 \left(\boldsymbol{s}_{t+1}, \boldsymbol{a}_t \right) \right) - Q_1(\boldsymbol{s}_t, \boldsymbol{a}_t) \right], \tag{3.21}$$

if Q_1 is updated, and

$$Q_2(\boldsymbol{s}_t, \boldsymbol{a}_t) = Q_2(\boldsymbol{s}_t, \boldsymbol{a}_t) + \gamma \left[r_t + \lambda Q_1 \left(s_{t+1}, \arg\max_a Q_2 \left(\boldsymbol{s}_{t+1}, \boldsymbol{a}_t \right) \right) - Q_2(\boldsymbol{s}_t, \boldsymbol{a}_t) \right], \tag{3.22}$$

if Q_2 is updated. This methodology mitigates bias by diversifying the update flow across two distinct tables, which increases the robustness and accuracy of the model.

3.3.6 Multi-Agent Q-Learning

In scenarios involving multiple decision-making entities or agents, such as DBSs, the standard Q-learning approach uses a central controller that observes the entire environmental state and decides individual actions for each agent. Using this model, the action space size increases exponentially with the number of agents, with a value of k^M. An alternative approach, as suggested by [50], is to adapt Q-learning for multistage decision processes with multiple controllers. This adaptation allows each agent to make decisions sequentially and independently within each decision epoch, based on both the global state and the actions previously chosen by other agents.

Specifically, the state observed by each agent j is modified to include decisions made by prior agents within the same epoch, that is

$$\boldsymbol{s}'_{j,t} = (\boldsymbol{s}_{j,t}, \boldsymbol{a}_t^1, ..., \boldsymbol{a}_t^{j-1}), \tag{3.23}$$

where a_t^i denotes the action taken by the i-th agent earlier in the same epoch. Each agent then decides its action based on this augmented state, using it own Q table, denoted as Q^j. This structure ensures that while the state space complexity increases with each additional agent due to the aggregated decisions (the space size is multiplied by $k^{(j-1)}$ for the j-th agent), the decision space of each agent remains fixed to k, which translates to decreased computational demands compared to a single controller model which would require making decisions across a space of k^M possibilities. This reformulation allows each agent to independently adapt and optimize its strategy with respect to both its immediate environment and the preceding decisions of its peers.

3.3.7 QoS Metrics for Q-Learning

This section presents the metrics used in designing the Q-learning states and reward.

Link reliability

The model assumes a minimum SINR threshold of Γ_{th} for a GN to be considered *served*. Link reliability is defined as the ratio of served users to total users. Using the number of served users per DBS outlined in (2.13) and the max-SINR binary association variable in (2.12), the link reliability per DBS is

$$\Phi_j(t) = \frac{N_j(t)}{\sum_{n=1}^{U} W_{j,n}(t)}, \tag{3.24}$$

where time indices are included since the metrics are obtained at different time instances during episodes of length T_{epoch}. Similarly, the link reliability for the entire system is

$$\Phi(t) = \frac{\sum_{j=1}^{M} N_j(t)}{U}. \tag{3.25}$$

Fairness

Ensuring the fairness of coverage in wireless communication systems is crucial for providing unbiased service quality, preventing user dissatisfaction, and maximizing overall network efficiency and reliability. In deploying DBSs, it's essential to consider fairness to avoid situations where only a subset of GNs in a specific area receives good SINR coverage, leaving others uncovered for long periods. To model system fairness, Jain's fairness index is used [58]. This metric utilizes the history of SINR measurement reports transmitted by a GN. It is assumed that SINR is reported every t_s, and fairness is calculated over a moving time

window of size $T_F \gg t_s$. A GN's coverage score at time t is based on its achieved SINR during the interval $[t - T_F, t]$ as

$$\mathrm{cov}_n(t) = \sum_{kt_s=t-T_F}^{t} \sum_{j=1}^{M} W_{j,n}(kt_s) H\left(\Gamma_{j,n}(kt_s) - \Gamma_{\mathrm{th}}\right), \tag{3.26}$$

which counts the number of time steps that GN received sufficient SINR. Using this coverage score, Jain's equation gives the fairness score per DBS j as

$$F_j(t) = \frac{(\sum_{n=1}^{U} W_{j,n}(t)\mathrm{cov}_n(t))^2}{(\sum_{n=1}^{U} W_{j,n}(t))(\sum_{n=1}^{U} W_{j,n}(t)\mathrm{cov}_n^2(t))}, \tag{3.27}$$

and for the entire system as

$$F(t) = \frac{(\sum_{n=1}^{U} \mathrm{cov}_n(t))^2}{U(\sum_{n=1}^{U} \mathrm{cov}_n^2(t))}. \tag{3.28}$$

3.3.8 Multi-Agent DBSs Q-Learning

The proposed RL mechanism involves each DBS as an agent that determines its next location at the start of each Q-learning epoch based on its own Q table. These decisions can be made autonomously by the DBS or directed by a central unit, such as a MBS, using the SINR history and performance metrics from Section 3.3.7. The decision process is sequential, ensuring each DBS is aware of prior decisions by other DBSs within the same epoch. At the epoch's end, rewards calculated from performance indicators update the Q tables.

Action space

The action space for each DBS includes all possible predetermined locations. The actions available for DBS j are defined by the set of locations, $\mathcal{K}_j$.

In the simulated scenario, the sets of possible locations are:

$$\begin{aligned}
\mathcal{K}_1 &= \{(100, 200),\ (300, 400)\},\\
\mathcal{K}_2 &= \{(700, 200),\ (500, 400)\},\\
\mathcal{K}_3 &= \{(100, 1000),\ (300, 800)\},\\
\mathcal{K}_4 &= \{(700, 1000),\ (500, 800)\},
\end{aligned} \tag{3.29}$$

where each location is defined by a tuple containing the x and y coordinates, respectively.

State space

The state observed by each DBS is composed of:

1. The current locations of all DBSs: $\{\mathbf{m}_j(t) \mid j \in \{1,\ldots,M\}\}$,
2. The quantized multiplication of reliability and fairness scores for each DBS: $\{\lfloor S_{\text{state}} F_j(t)\Phi_j(t)\rfloor \mid j \in \{1,\ldots,M\}\}$,
3. The actions taken by preceding DBSs, if any, during the current epoch.

where S_{state} controls the space size of the second component (e.g., setting $S_{\text{state}} = 10$ specifies a state space size of 10 since $F_j(t)\Phi_j(t) \in [0,1]$). These components are integrated into the state vector for DBS j as

$$s'_{j,t} = \left[\mathbf{m}_1(t),\ldots,\mathbf{m}_j(t),\lfloor S_{\text{state}} F_1(t)\Phi_1(t)\rfloor,\ldots,\lfloor S_{\text{state}} F_j(t)\Phi_j(t)\rfloor, a_t^1,\ldots,a_t^{j-1}\right]. \quad (3.30)$$

Reward

The reward is designed to encourage agents to improve the fairness and reliability metrics described in Section 3.3.7. Therefore, the author proposes the following value, proportional to the total fairness and reliability:

$$R = 100 \times F(t)\Phi(t), \quad (3.31)$$

where the reward is multiplied by a factor of 100 to speed up learning.

Fairness scores are calculated within an elapsed time window of T_F with a time step of t_s as explained in Section 3.3.7. A Q-learning epoch lasts T_{epoch}. State observation and action selection occur at the beginning of the epoch, while the reward is obtained at its end. T_{epoch} should be long enough for DBSs to reach their destinations by the end of the epoch. Algorithm 2 shows the pseudocode of the proposed Q-learning mechanism. Note that during testing the exploration probability ε is set to zero.

Algorithm 2 DBSs Placement with Q-learning

Input: γ, λ, ε, possible locations for each DBS, T_{epoch}

Initialization: initialize all values in Q_1 and Q_2 to zero

for number of cycles **do**

$\boldsymbol{s}'_{0,t} \leftarrow$ initial state without according to (3.30) without any prior DBS actions

for $j = 0$ to M **do**

Select action a_t^j maximizing $Q_1^j(\boldsymbol{s}'_{j,t}, \boldsymbol{a}_t) + Q_2^j(\boldsymbol{s}'_{j,t}, \boldsymbol{a}_t)$ with probability $1-\varepsilon$, otherwise choose randomly

Update state for next DBS by appending the selected action to the state vector: $\boldsymbol{s}'_{j+1,t} \leftarrow (\boldsymbol{s}'_{j,t}, a_t^j)$

end for

At end of T_{epoch}, compute reward using (3.31)

for $j = 0$ to M **do**

Update Q_1^j using (3.21) or Q_2^j using (3.22) with a 0.5 probability

end for

end for

This approach ensures that DBSs act based on up-to-date environmental feedback and thus adapt to changes in network conditions. This crucial for maintaining effective coverage and connectivity in complex wireless networks.

3.3.9 Simulations and discussion

Baseline scenarios

To show the gain in employing the proposed multi-agent Q-learning scheme, two benchmark solutions are utilized:

1. **Fixed deployment**: in this approach, the DBSs are placed in fixed locations with no change of locations. Thus, they consume the highest possible energy (see Section 2.5.1). The locations selected for this approach are: $\boldsymbol{m}_1 = (100, 200)$, $\boldsymbol{m}_2 = (700, 200)$, $\boldsymbol{m}_3 = (100, 1000)$, and $\boldsymbol{m}_4 = (700, 1000)$.

2. **Alternating deployment**: in this case, the DBSs always select the far location from their respective location sets in (3.29) to travel to in each epoch. Thus, this approach yields the lowest energy consumption since moving the drones consumes less energy than hovering.

Simulation settings

100 episodes of user trajectories are generated using the mobility model from [33] outlined in Section 2.1.2 within the environment shown in Figure 3.7. Each episode had a duration of $235 \times T_{\text{epoch}}$ and varied in the number of users U. The proposed algorithm's performance is compared with the previously described benchmark solutions. The model was trained for a total of 143,500 cycles. The DBSs were fixed at a height of $h = 60$ m because with a tabular Q-learning approach, the size of the state space should be controlled to prevent an excessively large number of training cycles for the Q table. However, if a function approximation, such as artificial neural network (ANN), is used, different locations with varying heights can be defined. The simulation parameters are summarized in Table 3.1.

Table 3.1 Simulation Parameters

Parameter	Value
DBS weight Ω and density of air ρ	39.22 N and 1.225 kg/m^2 [47]
Total area of DBS rotor disks A	0.78 m^2 [47]
Environment variables α and β (according to Section 2.2.1)	5.05 and 0.08
Carrier frequency f_c, bandwidth per user B_n, and DBS bandwidth B_D	2 GHz, 500 kHz, and 20 MHz
DBS RF transmission power P^T and receiver noise power σ_0^2	200 mW and -117 dBm
SINR threshold $\Gamma_{\text{th}}^{\text{dB}}$ and path loss threshold $L_{\max}$	10 dB and 100
Time step t_s, fairness time window T_F, and epoch duration T_{epoch}	2.5 s, 250 s, and 15 s
Discount ratio λ, learning rate α, and exploration probability ε	0.4, 0.4, and 0.2
State size S_{state} and number of possible locations per DBS k	5 and 2
DBSs' possible locations	see (3.29)

Convergence

In a Q-learning approach, if we take the average reward over N future cycles at each cycle and observe it increasing, it indicates that the learning algorithm is effectively improving its policy over time. An increasing average reward suggests that the agents optimize their actions to maximize long-term benefits. This trend validates the proposed Q-learning scheme by demonstrating convergence, as the agents consistently enhance their performance and adapt to the environment more efficiently. Figure 3.12 shows the average reward averaged

over 50 cycles in each cycle. We can observe that Q-learning starts to converge around 60,000 cycles. This indicates that the state space and reward function formulation enable intelligent decisions for moving DBSs based on the state, such that the achieved performance, indicated by the reward, increases as experience is accumulated.

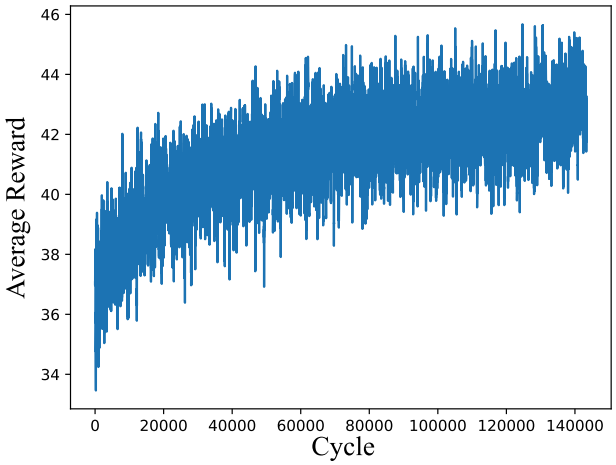


Figure 3.12 Average reward per cycle over 50 cycles obtained during training. The trend of increase proves convergence towards higher-yielding actions.

The size of the Q-table, proportional to the state and action spaces, determines the convergence speed. As more state-action pairs are defined, more sampling is required to obtain the optimal Q value in (3.18). Even with $M = 4$ and $k = 2$, the number of cycles needed was large. This highlights the need for using an ANN as a function approximator in a deep learning (DL) approach, allowing for further actions (e.g., different heights) and higher DBSs count. Nevertheless, employing the tabular approach and observing convergence confirms the sanity of the designed framework.

Link reliability, fairness, and rewards

The author simulated 100 episodes, each with 3525 seconds of testing (235 cycles $\times 15$ seconds) with $\varepsilon = 0$. Figure 3.13 shows the achieved reward, total link reliability, and total fairness score per cycle averaged across episodes for different GNs counts. The proposed algorithm substantially outperforms benchmark methods in terms of fairness and reward. The fixed deployment performs similarly to the Q-learning method in terms of reliability since the DBSs are always placed to maximize distances and minimize interference, leading to increased link reliability. The alternating solution improves fairness compared to the fixed deployment, as DBSs change positions and transmit higher power to different user

subsets at different times. However, the intelligent planning of movement using multi-agent Q-learning achieves superior fairness compared to both alternating and fixed schemes. The results indicate that increasing the number of GNs does not significantly affect the outcomes, demonstrating the scalability of the proposed framework.

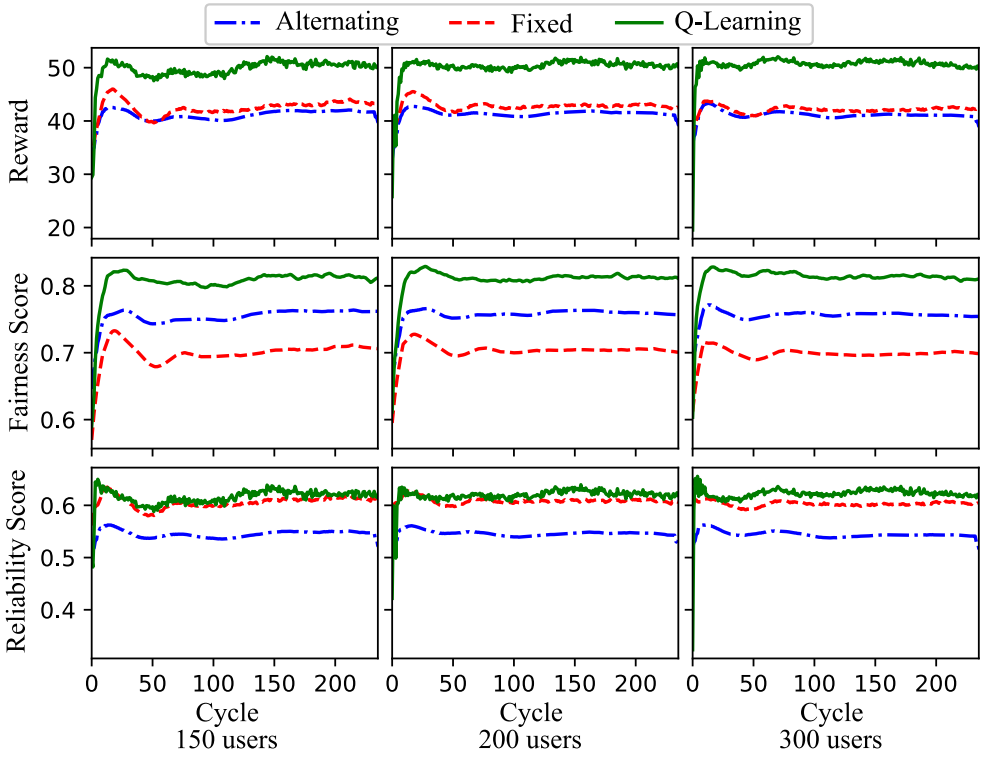


Figure 3.13 Measured QoS values per testing cycle averaged across 100 episodes for different numbers of GNs, U. The approach is independent of U and scales well with an increasing number of GNs. The proposed scheme outperforms the two benchmark solutions in terms of reward and fairness. The fixed benchmark solution yields similar reliability results since DBSs do not come near to each other, but performs worse in terms of fairness.

Aerodynamic energy consumption

Figure 3.14 shows the total aerodynamic energy consumed across all cycles averaged across 100 episodes. The alternating solution yields the lowest consumption, forcing DBSs to change locations at each cycle. The fixed benchmark solution results in the highest power consumption, with an increase of 107% compared to the alternating case. The proposed solution performs in between, with a 65% increase relative to the alternating scenario and a 20% decrease compared to the fixed solution. Again, changing the number of GNs does not affect the performance in terms of energy.

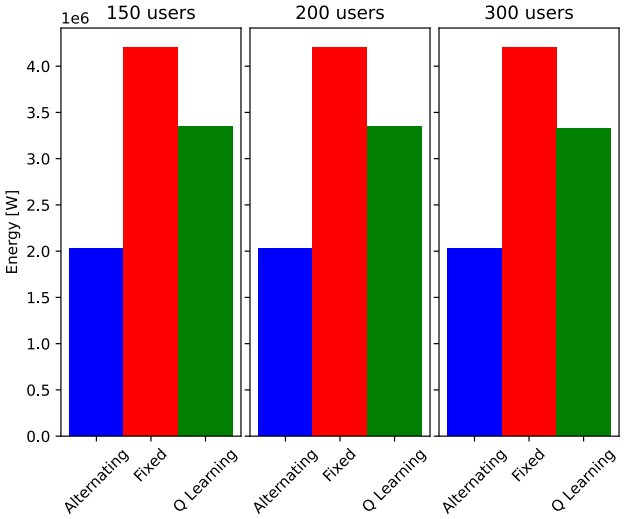


Figure 3.14 Total energy consumption during 235 cycles, averaged across 100 episodes. The fixed solutions, with DBSs hovering in one spot, consume the most energy, while the alternating solution consumes the least. The proposed RL algorithm falls in between, showing that intelligent movement of DBSs can also result in power savings.

Mean SINR per episode

For each GN, the mean received SINR value is obtained during each episode, averaged over all measurements taken every t_s within the 235 simulated T_{epoch} cycles. That is, $235 \times 100 \times U$ mean SINR values are obtained for each plot. Figure 3.15 shows the CDF of these mean values. We observe that naive movement of DBSs degrades the received SINR due to increased interference. Conversely, intelligent movement decisions perform slightly better than those in fixed scenarios, which incur the least interference. This indicates that after the learning process, DBSs are positioned to minimize interference, even without knowledge of GNs locations, relying only on measurement history. Again, performance is stable across even as the number of GNs increases.

FDMA achieved throughput

In accordance with the scope defined in Section 1.5.1, the author has so far neglected the actual allocation of frequency resources to the GNs and focused only on the measured SINR, assuming fixed allocated channels as described in Section 2.2.1. However, to study the effects on load balancing of cells compared to benchmark schemes, additional results are presented. Assuming GNs are associated with the nearest DBS, and at each measurement instance (nt_s), the 20 MHz bandwidth of DBS j is divided by the number of its served users to obtain the

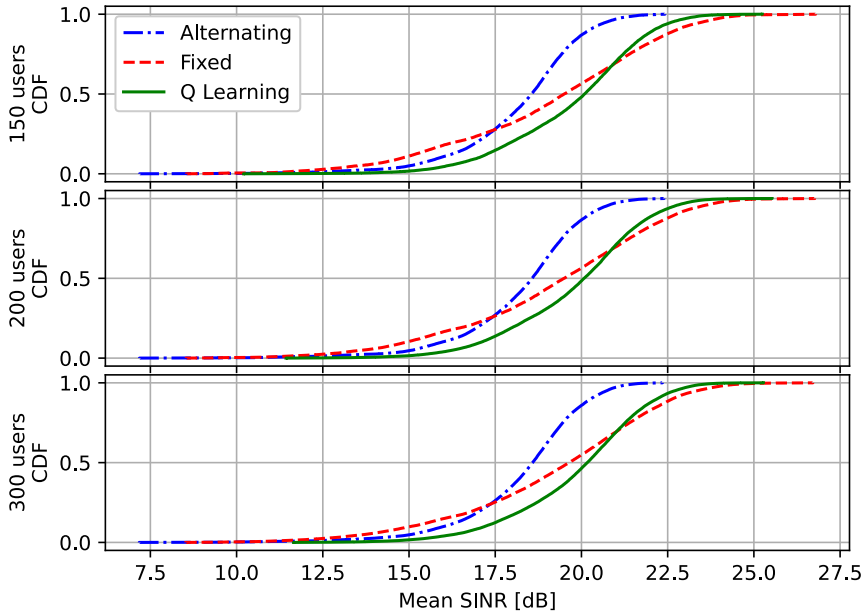


Figure 3.15 CDF plots of mean GN SINR across the simulation period averaged over the 100 simulated episodes. The alternating solutions perform the worst due to increased interference.

corresponding B_n settings. Then, the throughput of each GN is calculated using (2.8), (2.10), and (2.11). The boxplots in Figure 3.16 illustrate these throughput values.

The first row shows the distribution of measured throughput values for all users. On average, the proposed solution provides higher throughput values. The fixed case, despite similar mean SINR values (see Figure 3.15), results in lower mean throughput values, indicating reduced efficiency in resource usage, and gain when moving cells. The second and third rows show the minimum and maximum throughput values observed by a GN within the entire episode. As the number of users increases, all three methods exhibit similar minimum throughput values. However, the maximum throughput achieved by the proposed algorithm outperforms both benchmark schemes for the highest number of GNs (300) and shows similar performance otherwise. This is expected, as the algorithm aims for fairness of coverage, ensuring that all GNs receive good coverage periodically.

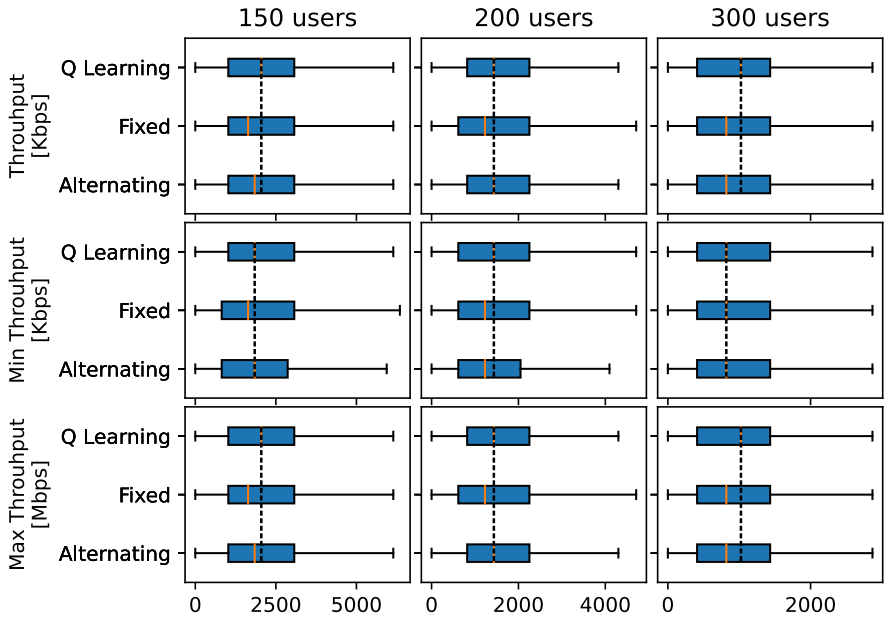


Figure 3.16 Boxplots of throughput values. The first row is based on all measured throughput values for all users, the second row is based on the minimum measured throughput value throughout each episode, and the final row is based on the maximum throughput value throughout each episode. The dashed black line marks the metric mean value of the Q-learning approach. In most cases, the proposed mechanism outperforms the benchmark schemes.

3.4 Placement with Beamforming to Service Vehicular Users in Streets

The 5G New Radio (NR) specification, starting with its 16th 3GPP release, has introduced a dedicated NR-V2X air-interface that enables vehicles communication with other vehicles or infrastructure endpoints such as roadside units or cloud endpoints like application servers. This provides the technological foundation upon which the research community is now building mechanisms for reliable cooperative driving, real-time traffic optimization, and fully autonomous cars [72].

The reliability of these networks is crucial. Failure to deliver packets might result in a missing collision warning. Moreover, high reliability can enable autonomous driving and boost traffic efficiency. Solutions should be designed for cases where more channels might be needed. For example, some areas might suffer from bandwidth congestion due to high traffic density or limited bandwidth availability. These concerns are amplified by the high

demand imposed by vehicular services such as shared sensing, which transmits camera feeds, high-definition mapping, or software updates.

Again, to boost reliability, DBSs can be deployed in such scenarios or in emergency situations where infrastructure is lacking. This section targets the problem of mapping the reliability of DBSs deployment into a measurable metric that can be used to evaluate a deployment's performance. The targeted metric is the access probability, which quantifies the chance of a vehicular user to get serviced by the coverage provided by DBSs, given the density of vehicles in an area.

The work presented below builds upon the analysis in [73], where an expression for the access probability was derived for vehicles moving in a straight line under a given beam. In this thesis, a slightly modified model is proposed, which yields more accurate results, as presented below.

3.4.1 System Model

An illustration of the considered model is shown in Figure 3.17, where a DBS provides coverage to vehicular users within a portion of a lane via a directional beam. Vehicles are assumed to be distributed with a one-dimensional PPP, characterized by the probability mass function (PMF):

$$p_k(k) = \frac{\bar{k}^k \cdot e^{-\bar{k}}}{k!}, \tag{3.32}$$

where k denotes the number of vehicles, with mean $\bar{k} = \lambda_v L_b$. Here, L_b is the beam-covered lane length, and λ_v is the vehicle density. The beam's width is proportional to the azimuth beamwidth angle φ_b, and its length is similarly proportional to the elevation beamwidth angle θ_b.

Following the simplified model in [73], the beamforming gain is given by:

$$G_b(\varphi_b, \theta_b) = \frac{41000}{\varphi_b^\circ \times \theta_b^\circ}, \tag{3.33}$$

indicating that narrower beams yield higher gain.

Considering the downlink direction, the received power at a vehicular user is expressed in dB as:

$$P_R = P_T + G_t + G_b(\varphi_b, \theta_b) - L_b^{[\mathrm{dB}]}, \tag{3.34}$$

where P_T is the transmit power, G_t is the gain of the transmitting antenna, and $L_b^{[\mathrm{dB}]}$ is the path loss defined as:

$$L_b^{[\mathrm{dB}]} = \mathrm{FSPL} + s, \tag{3.35}$$

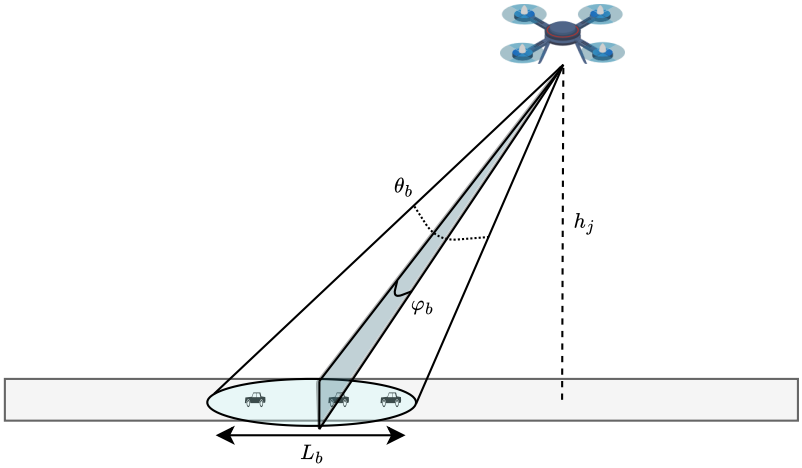


Figure 3.17 Illustration of a general scenario where a DBS provides coverage to vehicular users via a directional beam. The beam is characterized by its width, denoted by the beamwidth azimuth angle φ_b, and its length, which is proportional to the beamwidth elevation angle θ_b.

with FSPL being the free-space path loss (see 2.5) and s a log-normal shadowing sample.

Assuming a minimum received power threshold $P_{r_{\min}}$, the outage probability is given by:

$$\mathcal{P}_{\text{out}} = \mathcal{P}(P_r + s < P_{r_{\min}}) = 1 - \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{P_{r_{\min}} - P_r}{\sqrt{2} \cdot \sigma_s}\right), \tag{3.36}$$

where σ_s is the standard deviation of the log-normal shadowing distribution, and $\operatorname{erfc}(\cdot)$ is the complementary error function.

In addition to meeting the minimum power requirement $P_{r_{\min}}$, each vehicle must be allocated sufficient bandwidth to meet its throughput demand. Assuming equal bandwidth allocation among vehicles within the same beam, the maximum number of supported vehicles is:

$$K_0 = \frac{B_0}{B_n^*}, \tag{3.37}$$

where B_0 is the total bandwidth, and B_n^* is the bandwidth required to satisfy the minimum throughput, assuming the user is served at $P_r = P_{r_{\min}}$. While this simplification ignores power variations between users that would vary the bandwidth assignment in optimal scenarios, it nevertheless enables a closed-form expression for the access probability, as described below.

3.4.2 Old Access Probability Model

In [73], the probability of a user achieving successful connectivity under a given beam is expressed as:

$$\mathcal{P}_A^{\text{old}} = (1 - \mathcal{P}_{\text{out}}) \left(1 - \mathcal{P}_{\overline{\text{res}}}^{\text{old}}\right), \tag{3.38}$$

where $\mathcal{P}_{\text{out}}$ is defined in (3.36), and $\mathcal{P}_{\overline{\text{res}}}^{\text{old}}$ denotes the probability of insufficient resources to serve all covered users, computed in [73] as:

$$\mathcal{P}_{\overline{\text{res}}}^{\text{old}} = \sum_{k=\lceil K_0 \rceil}^{\infty} \frac{\overline{k}_b^k \cdot e^{-\overline{k}_b}}{k!}, \tag{3.39}$$

where $\overline{k}_b$ is the average number of vehicles within the beam that satisfy the power threshold $P_{r_{\min}}$. This is obtained as $\overline{k}_b = \left(1 - \mathcal{P}_{\text{out}}^{\text{mid}}\right) \bar{k}$, using a representative outage probability evaluated for a vehicle positioned at the beam midpoint.

3.4.3 Novel Access Probability Model

Below is a more accurate access probability model, building on the earlier formulation. It incorporates modifications that improve the accuracy of simulation results.

First, the author considers how to compute the probability of having n users with sufficient received power out of k total users. This will be used in the calculation of the new access probability described below. Given an average outage probability $\overline{\mathcal{P}}_{\text{out}}$ (the new model uses average instead of midpoint value), the probability of having n covered users is given by the binomial distribution:

$$\mathcal{P}_{\text{cov}}(n \mid k) = \binom{k}{n} \left(1 - \overline{\mathcal{P}}_{\text{out}}\right)^n \overline{\mathcal{P}}_{\text{out}}^{k-n}. \tag{3.40}$$

Next, assume there are $k > K_0$ users. Insufficient resources occur only if more than K_0 of them are covered, i.e., have sufficient $P_{r_{\min}}$. If we condition on a particular vehicle being covered and seek the probability that it cannot be served due to limited resources, then this probability can be obtained as:

$$\mathcal{P}_{\overline{\text{res}}}^{\text{new}}(k) = \sum_{n=\lfloor K_0+1 \rfloor}^{k} \mathcal{P}_{\text{cov}}(n-1 \mid k-1). \tag{3.41}$$

This represents the probability of insufficient resources when k users are present, given that the considered vehicle under question is already covered and therefore excluded from the combinations.

To obtain the overall probability of lack of resources, a sum over all possible values of k is obtained, using (3.32) and (3.41):

$$\mathcal{P}_{\overline{\text{res}}}^{\text{new}} = \sum_{k=\lfloor K_0+1 \rfloor}^{\infty} p_k(k)\, \mathcal{P}_{\overline{\text{res}}}^{\text{new}}(k). \tag{3.42}$$

Finally, a vehicle successfully accesses the network if it receives sufficient power and resources. The overall access probability is then given by:

$$\mathcal{P}_A^{\text{new}} = 1 - \mathcal{P}_{\overline{\text{res}}}^{\text{new}} - \overline{\mathcal{P}}_{\text{out}}. \tag{3.43}$$

Note that, unlike the earlier model, which relied on $\mathcal{P}^{\text{mid}}$out as a single representative estimate, the proposed model uses the average value $\overline{\mathcal{P}}_{\text{out}}$, obtained by sampling $\mathcal{P}_{\text{out}}$ at successive points along the entire beam and averaging the results.

3.4.4 DBS Placement and Beam-Footprint Association

Using the access probability model developed above, we evaluate, for every candidate DBS site i and every beam footprint location j, the access probability $\mathcal{P}_A(i,j)$. A link (i,j) is considered feasible if (i) a LOS path exists between site i and footprint j, and (ii) $\mathcal{P}_A(i,j) \geq \mathcal{P}_{A,\min}$, where $\mathcal{P}_{A,\min}$ is a design threshold that enforces a minimum reliability level. Let

$$a_{ij} = \begin{cases} 1, & \text{if LOS holds and } \mathcal{P}_A(i,j) \geq \mathcal{P}_{A,\min}, \\ 0, & \text{otherwise,} \end{cases} \tag{3.44}$$

and define the candidate DBS location set $\mathcal{I}$ (indexed by i) and the set of beam footprints $\mathcal{J}$ (indexed by j). Each active DBS can simultaneously serve at most N_{beams} footprints.

We select which sites to activate and how to associate footprints through the binary decision variables

$$y_i \in \{0,1\} \;\; (i \in \mathcal{I}), \qquad x_{ij} \in \{0,1\} \;\; (i \in \mathcal{I},\, j \in \mathcal{J}), \tag{3.45}$$

where $y_i = 1$ if site i is activated, and $x_{ij} = 1$ if footprint j is assigned to site i.

Furthermore, we define a MILP which seeks the minimum number of active DBSs that (i) cover all footprints and (ii) respect the per-DBS beam capacity, while discarding infeasible

links:

$$\min_{\{x_{ij},y_i\}} \sum_{i\in\mathcal{I}} y_i \tag{3.46}$$

$$\text{s.t.} \sum_{i\in\mathcal{I}} x_{ij} = 1, \qquad \forall j \in \mathcal{J} \tag{C1}$$

$$\sum_{j\in\mathcal{J}} x_{ij} \le N_{\text{beams}}, y_i, \qquad \forall i \in \mathcal{I} \tag{C2}$$

$$\sum_{j\in\mathcal{J}} x_{ij} \ge y_i, \qquad \forall i \in \mathcal{I} \tag{C3}$$

$$x_{ij} \le a_{ij}, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{C4}$$

$$x_{ij} \in \{0,1\},\ y_i \in \{0,1\}, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J}. \tag{C5}$$

where (C1) assigns each footprint to exactly one active site, ensuring full coverage; (C2) enforces the per-site beam limit; (C3) prevents opening an idle site; and (C4) forbids associations without LOS or with $\mathcal{P}_A$ below the threshold. Problem (3.46) is a capacitated facility-location/assignment MILP where opening a facility has unit cost.

This formulation leverages the reliability metric to make deployment decisions. By thresholding $\mathcal{P}_A$ into a_{ij}, the model admits only associations that meet the target reliability, and then selects the smallest set of DBSs capable of serving all footprints under beam-capacity constraints. The outcome is (i) a subset of activated sites $\{i : y_i = 1\}$ and (ii) a feasible association plan $\{x_{ij} = 1\}$ suitable for subsequent scheduling.

3.5 Simulations and Discussion

3.5.1 Old vs. New Access Probability Model

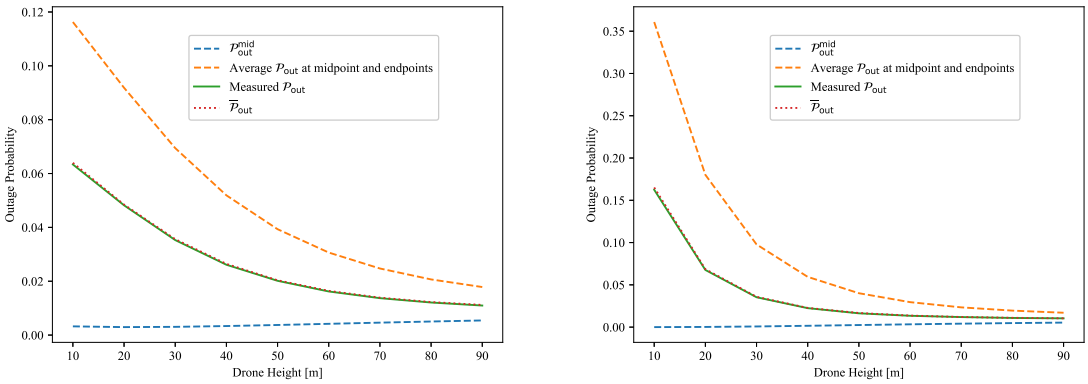
To obtain the results below, a single beam was simulated based on its location and that of the serving DBS. Vehicles were generated by sampling their inter-arrival times from an exponential distribution with the appropriate rate, yielding the required density $\bar{k}$. At each simulation time step t_s, the channel to each vehicle was sampled according to the model described in (3.34). The number of vehicles receiving sufficient power then determined their success in obtaining access to the network, as described earlier. The simulation parameters are listed in Table 3.2.

Firstly, we show the gain in terms of accuracy when the steps-average estimate of outage probability, $\overline{\mathcal{P}}_{\text{out}}$, is used instead of the mid-point estimate, $\mathcal{P}_{\text{out}}^{\text{mid}}$, used in [73]. Figure 3.18 shows the results obtained from two different simulation scenarios, with each having the DBS

Table 3.2 Simulation Parameters

Parameter	Value
Beam length L_b	146.8 m
Shadowing standard deviation σ_s	4
Required minimum received power	−52 dBm
Maximum number of supported vehicles K_0	5.4
Transmission power P_T	5 dBm
Carrier frequency f_c	27.5 GHz
Poisson density λ_v	0.6

placed at different instances along the street axis. Clearly, obtaining an step-averaged estimate yields more accurate results, while the mid point estimate fails to include the variations in outage probability due to the path loss across changes across the street.



(a) DBS location is at $\frac{L_b}{4}$.

(b) DBS is at $\frac{L_b}{2}$.

Figure 3.18 $\overline{\mathcal{P}}_{\text{out}}$ results in far more accurate results in comparison with $\mathcal{P}^{\text{mid}}$out.

Then, Figure 3.19 compares the old and novel models in terms of accuracy. In this case, both models use the step-averaged $\overline{\mathcal{P}}_{\text{out}}$. As shown, the novel model yields results much closer to the simulation measurements, showing higher accuracy than the model in [73].

3.5.2 Case Study: Selecting Beam-Footprint Associations in Bologna

We solve the proposed MILP for given a_{ij} and $\mathcal{P}_A(i,j)$ and compare it against a greedy baseline. The quantities a_{ij} and $\mathcal{P}_A(i,j)$ resemble values obtained from randomly placing candidate DBS sites in Bologna, selecting the main streets, partitioning them into equal-length footprints, and, using the novel access probability model, thus computing $\mathcal{P}_A(i,j)$

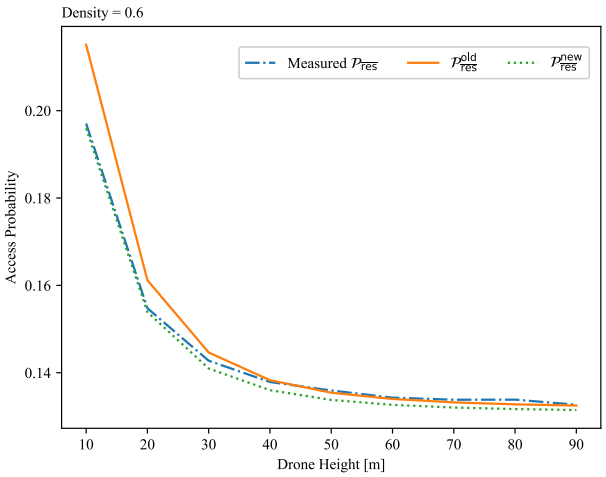


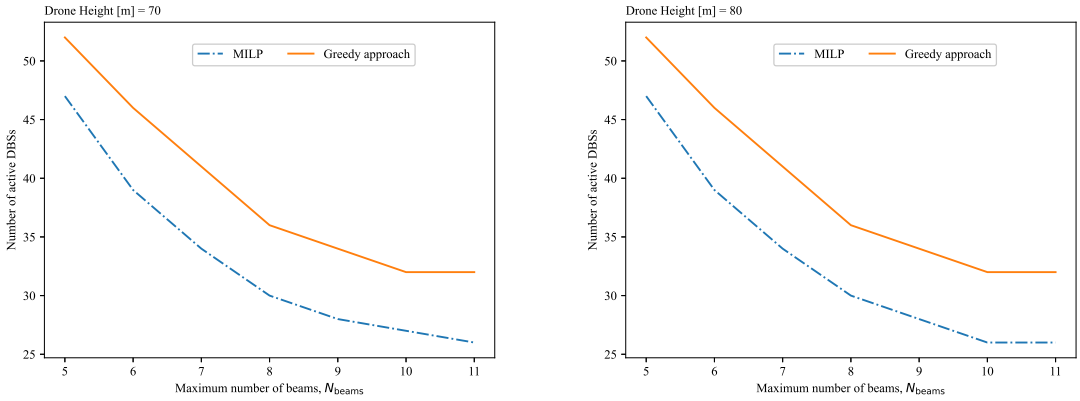
Figure 3.19 Comparison of the two access probability models and measured results.

and the feasibility mask a_{ij}. For calculation of LOS feasbility, a similar approach to that described in Chapter 5 can be utilized.

The greedy baseline works as follows. It orders footprints in ascending order by the number of feasible serving sites (the count of i with $a_{ij} = 1$) and iteratively assigns those with the fewest options first. At each step, it prefers beams from already activated DBS locations to avoid underutilization. If none are available, it activates a new DBS. The process continues until all footprints are covered, while enforcing that no DBS serves more than N_{beams} footprints. Experiments are conducted for $N_{\text{beams}} \in \{5, 6, 7, 8, 9, 10, 11\}$ and DBS heights of 70 m and 80 m. The reliability threshold is $\mathcal{P}_{A,\min} = 0.5$. The MILP is solved using CPLEX on Python.

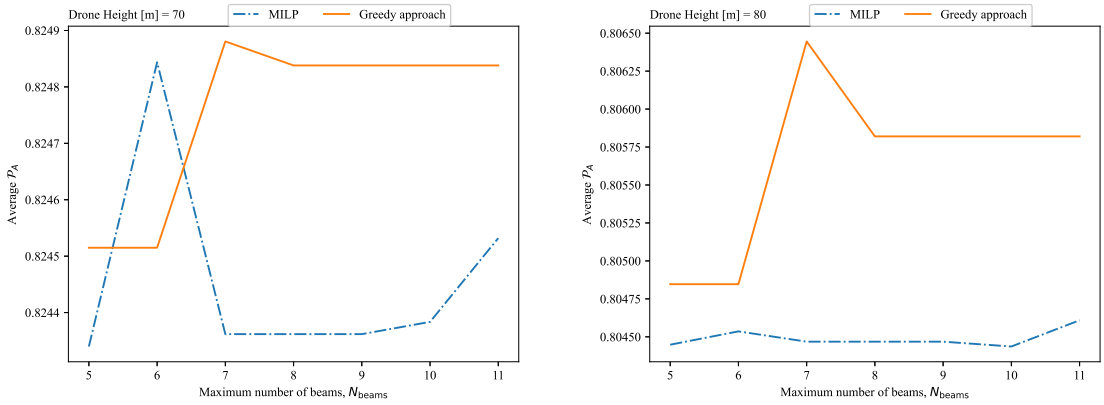
The first results in Figure 3.20 compare the number of active DBSs achieved by the MILP vs. the greedy approach. Note that changing the height of the DBSs seems to not change the results, and clearly, the MILP approach is superior in terms of reducing the number of active DBSs.

The second set of results in Figure 3.21 compares the minimum access probability achieved by the MILP with that of the greedy approach. Although the outcomes are very similar, the greedy method appears to slightly outperform the MILP in most cases, with differences not exceeding 0.001.



(a) Height $h = 70$ m. (b) Height $h = 80$ m.

Figure 3.20 Greedy vs. MILP approach for activating beams. Employing the MILP reduces the number of active DBSs.



(a) Height $h = 70$ m. (b) Height $h = 80$ m.

Figure 3.21 Greedy vs. MILP approach for activating beams. The minimum access probability results are similar, with small improvement observed when employing the greedy approach.

3.6 Conclusion

In this chapter, the problem of fronthaul-aware placement of DBSs is approached under two different scenarios: first, assuming knowledge of the exact coordinates of GNs; and second, assuming no knowledge of GNs locations. For the first case, a novel clustering method, SINR-EM, is introduced, which accounts for varying transmission power settings and interference among DBSs, using a modified EM framework guided by SINR-based similarity metrics. Simulation results confirmed its superiority over traditional clustering methods, particularly

in reducing interference and improving user link quality and thus the energy efficiency of the system. For the second case, the author addressed the placement problem by proposing a multi-agent Q-learning framework. This model-free approach enables DBSs to learn optimal placement policies based on historical channel measurements and fairness-aware quality of service (QOS) metrics. The simulation results demonstrated improved system fairness and energy efficiency compared to benchmark methods, while maintaining scalability with user density.

Finally, the chapter presented a mathematical model for mapping the estimated access probability of vehicular users served by a beam from the DBS. Building on the work in [73], the model introduces corrections and a mathematical reformulation. Simulation results confirm its superior accuracy compared to the earlier approach. The author also formulated an MILP showing how the model can be utilized to select DBS deployment and beam-footprint associations, while minimizing the number of active DBSs.

All solutions in this chapter focus solely on the fronthaul link and neglect backhaul provisioning. Alternatively, in the next chapter, drones are placed while considering backhaul links optimization aspects only.

Chapter 4

Optimizing Transmission Power

In this chapter, a transmission power control scheme for multiple DBSs deployment is presented. Using an adapted Monte Carlo tree search (MCTS) algorithm [74], the transmission power settings are determined for the deployed DBSs. Using this scheme, the number of required DBSs can also be determined, as some transmission power settings can be set to zero, thereby removing the DBS from the network.

The algorithm considers energy efficiency and link reliability metrics defined in (2.16) and (3.3.7). The problem is formulated as a game tree where each node represents a different combination of transmission power settings. The tree is solved using an adapted MCTS approach [74], which iteratively builds a search tree, performs random simulations, and updates the tree with outcomes to balance exploration and exploitation, refining optimal moves. MCTS is a type of RL algorithm [70]. The author simulates the performance of this algorithm when combined with the SINR-EM placement algorithm presented in Section 3.2. The performance is also compared with another algorithm that has received considerable attention in the literature. This algorithm employs particle swarm optimization algorithm (PSO) to determine the 3D locations of the DBSs. In contrast, the proposed MCTS scheme optimizes only the 2D locations and transmission power. Therefore, the author provides a performance comparison between optimizing the heights (PSO) and optimizing the transmission power (MCTS).

The contributions in this chapter were published in:

1. Janji, S. & Kliks, A. *Drone Base Stations Transmission Power Control and Localization*. In: Mobile and Ubiquitous Systems: Computing, Networking and Services, LNICS, vol. 492, Springer, pp. 356–377, 2023. https://link.springer.com/10.1007/978-3-031-34776-4_19

4.1 Related Work

In [75], the authors addressed the simultaneous wireless information and power transfer (SWIPT) problem, aiming to minimize the energy consumption of a single DBS while satisfying user data rate requirements. The problem is decoupled into optimizing the DBS's transmission power and position, and the SWIPT time and power splitting ratios. The iterative optimization process achieves significant energy savings and maintains user QoS. The centroid method, combined with gradient descent, is used to reduce problem complexity and solve for decision variables. For a sensor network assisted by a single DBS sharing the spectrum with all sensors, [76] addresses interference management by optimizing transmit power and DBS trajectory to maximize sum throughput. The problem is divided into transmit power control and trajectory planning, using difference of convex functions programming and successive convex approximation to find suboptimal solutions. In [77], the authors aim to maximize the secrecy rate by jointly optimizing UAV trajectory and transmission power in the presence of eavesdroppers. The problem is formulated as a non-convex optimization and solved using alternating optimization. [78] focuses on optimizing UAV positioning and transmission power to enhance communication reliability and throughput. The problem is approached by decoupling UAV positioning and power control into separate tasks, employing heuristic and optimization techniques for iterative improvement. These works consider a single DBS scenario only and do not address the complexity arising from interference between multiple deployed DBSs and how to control their power accordingly.

4.2 Monte-Carlo Tree Search MCTS

In its simplest form, MCTS incrementally builds an asymmetric tree where each node defines a unique decision at a given state. At each iteration, the algorithm begins at the root node and progresses to a specific node through a series of selections that conform to a certain policy, which balances between exploration of new states and exploitation of previous results. Next, the selected node is expanded by adding a child node to the tree. Starting from this child node, a simulation (or playout) is run until a terminal state is reached. In the context of games, this can be a *win* or *loss*. Finally, the result is back-propagated from the terminal state through its ancestors to the root node by updating the states of those nodes [74].

4.3 Transmission Power MCTS (TP-MCTS)

An adapted MCTS algorithm is proposed to determine each DBS transmission power. The algorithm is denoted as transmission power Monte Carlo tree search (TP-MCTS). In TP-MCTS each node is uniquely defined by a set of transmission power settings for all DBSs. Starting from an initial setting of maximum power for all DBSs (i.e., $\boldsymbol{P}_T = \left[P_{T_1} = P_{\max}, .., P_{T_M} = P_{\max}\right]$) a root node is defined. A child node is created by decreasing the transmission power of one DBS by a predefined step of P_Δ (e.g., $\boldsymbol{P}_T = [P_{\max}, P_{\max} - P_\Delta, \ldots, P_{\max}]$). For each selected node, SINR-EM is performed given the specified transmission powers, and two figures of merit are obtained as a result: the first is the number of served users, N_{node}, given by (2.14); and the second is the achieved energy efficiency, ζ_{node}, given by (2.16). Both are used to obtain the score (reward) of each node. Let $N_{\text{th}} = \Phi U$ be the minimum required number of served users achieving an SINR value greater than or equal to Γ_{th}, where Φ is the desired percentage of served users (or link reliability). Then, the score of each node in the i-th iteration is obtained as

$$S_{\text{node}}^{(i)} = \begin{cases} \frac{N_{\text{node}}^{(i)}}{U} & \text{if } N_{\text{node}}^{(i)} < N_{\text{th}}. \\ \Phi + \frac{\zeta_{\text{node}}^{(i)}}{B_n} & \text{otherwise.} \end{cases} \tag{4.1}$$

If N_{th} is achieved, then the score would also include a value proportional to the achieved energy efficiency, independently of the assigned bandwidth, to search for solutions that further optimize the system's energy efficiency, since the score guides the tree search as will be explained later below.

Figures 4.1a to 4.1d illustrate an example of TP-MCTS for four iterations. Each node is defined by the set of specified transmission powers below it, and its state contains the score and number of visits. Note that the number of visits is incremented only when a child of that node is visited through it, not when it is selected. For the sake of demonstration, it is assumed that the root node initially has a score of 0.6 obtained from the deployment results of SINR-EM with the initial maximum transmission power settings. The algorithm proceeds as follows:

- **Iteration 1**: The child node that reduces the transmission power of the first drone is selected (the selection criterion will be presented later). After performing SINR-EM, a result of 0.65 is achieved, which is higher than the node's sole ancestor (i.e., the root) score, so a *win* has occurred and the result is back-propagated upwards.

- **Iteration 2**: The root node score is updated with the resulting *win* from its child. When a *win* occurs, all of the winning node ancestors' scores are updated after receiving

the score (see (4.1)) achieved by the winning child. The new scores are calculated according to the cumulative average

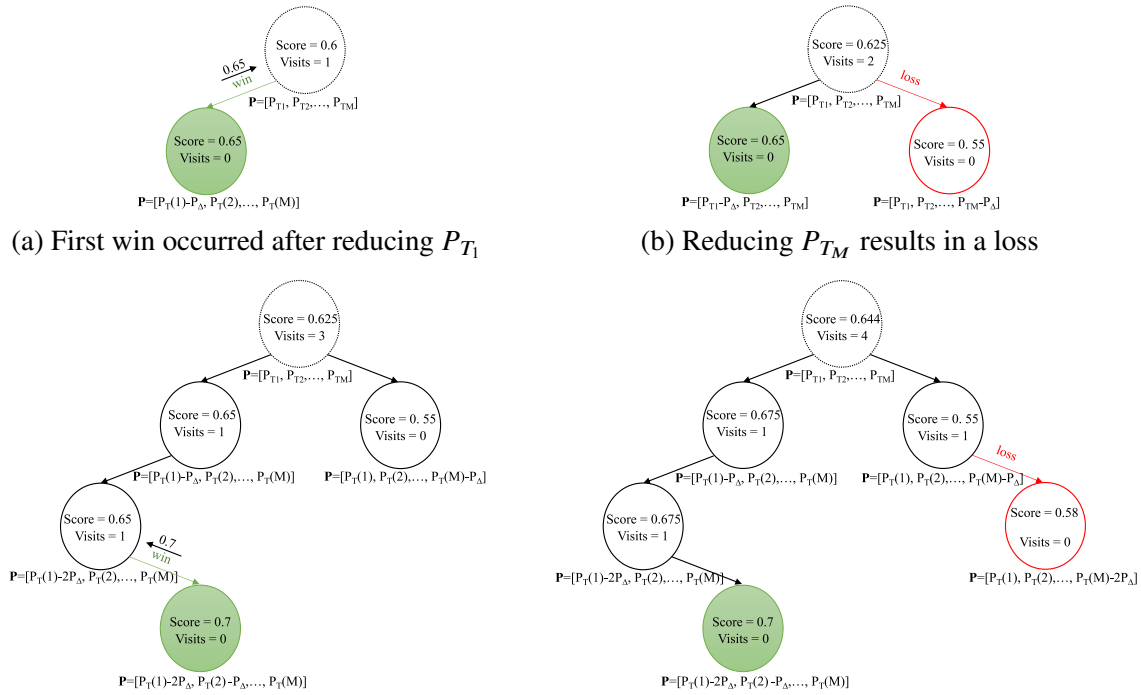
$$S^{i+1}_{\text{node}} = S^{(}_{\text{node}}i) + \left(\frac{S^{(}_{\text{win}}i) - S^{(}i)}{V^{(}i)_{\text{node}} + 1} \right), \tag{4.2}$$

where $V^{(}i)_{\text{node}}$ is the visits count and $S^{(}_{\text{win}}i)$ is the winning child node's score. The next selected node reduces the transmission power of DBS M and results in a score lower than its highest ancestor score (the root in this case) and therefore a *loss* occurs. The score is not back-propagated in the case of *loss*.

- **Iteration 3**: The child node with the reduced transmission power for the first DBS is selected again. Given that this node was selected before then, it is not considered a *win* and therefore the algorithm continues. That is, a *win* or *loss* occurs only when a node is selected for the first time. Next, a further child node is created by again reducing the transmission power of the first drone ($j = 1$), and SINR-EM results in a score of 0.65. Since this score is equal to the highest ancestor score (0.65 belonging to the parent in this case), then again no *win* or *loss* occurs. Finally, another child node is selected, which reduces the transmission power of the second drone ($j = 2$), and the obtained score is 0.7, which is higher than 0.65, and therefore a *win* has occurred, and the result is back propagated. The optimal state (shaded) is the one that has so far achieved the highest score during creation.

- **Iteration 4**: Assuming that the losing node of the second iteration is selected again, we observe that no loss occurs. Next, a further child node is created by an additional decrease of the last DBS power ($j = M$), and the achieved score is 0.58. Although the score is higher than its parent node, a loss occurs because the highest ancestor score is 0.644, belonging to the root node, which is higher than the score achieved by this child.

In TP-MCTS the selection, expansion, and simulation steps of the standard MCTS are lumped together, and at each iteration, the algorithm progresses by selecting child nodes until reaching a terminal node which is selected for the first time such that it terminates with a *win* or *loss* if the achieved score of the considered node is higher or lower respectively than the highest score in the traversed nodes. The score is back-propagated only in the case of *win*.



(a) First win occurred after reducing P_{T_1}

(b) Reducing P_{T_M} results in a loss

(c) Second win after reducing P_{T_2}

(d) Second loss after reducing P_{T_M} again

Figure 4.1 MCTS example. Note how the ancestor scores are updated after each iteration.

4.3.1 Selection Criterion

To select the next node, the scores of all possible child nodes are considered, while the missing scores of the nodes that are not yet created are substituted by the parent score (i.e., current). The selected child node is the one with the maximum upper confidence bound score (UCB) value given by [79]

$$\mathrm{UCB}_{child} = S_{child} + K_e \sqrt{\frac{\ln(visits_{parent})}{visits_{child} + 1}}, \tag{4.3}$$

where K_e is an arbitrary constant controlling the level of exploration.

Algorithm 3 shows the object-oriented programming (OOP) pseudocode for TP-MCTS, which is implemented recursively. Each node is an object (line 7) which can be simulated by invoking the SIMULATE method (line 12). Each child receives its ancestor's highest score to check for a *win* or *loss* after performing SINR-EM with the given transmission power $\boldsymbol{P}_T$ and set of users $\mathcal{U}$. Starting with the *root* node, and for a given number of max iterations ($iteration_{\max}$), the SIMULATE method is invoked. On the first call to each node's SIMULATE method, the score is obtained after performing SINR-EM and it is compared to the global highest score to keep track of the optimal simulated state (line 16). Then, each

possible child of the current node is instantiated by decreasing the transmission power of a uniquely selected drone, and the score of each child is initialized to the parent's achieved score as mentioned previously. Then, the achieved score is compared to the highest ancestor score, $S_{ancestor}$, which is the highest score between the nodes on the path from the parent of the current node to the root. In the case of a *win* (line 26), the achieved score is returned; in the case of a *loss* 0 is returned. If it is not the first simulation of the node or no *win* or *loss* occurred, the algorithm proceeds by selecting a child of the current node according to the UCB policy (see (4.3)) and invoking its SIMULATE method. Whenever a positive score resulting from a win is back-propagated from the child simulation, the parent node updates its score according to (4.2).

4.3.2 Time-based Updates

Assuming that DBSs are deployed to serve non-stationary users, the mobility of drones can be utilized to adapt to the movement of users and periodically update the locations of DBSs so that the overall performance of the system is improved. Instead of initializing the TP-MCTS tree at each update interval and performing a substantial number of iterations to approach the optimal solution, we can make use of the experience embedded in the current tree structure. To proceed further, the following assumption is made: after a duration of t_{update}, the new optimal solution lies in the vicinity of the previously found solution. That is if t_{update} is small enough relative to the speed of movement of users, then the existing branches weights (scores) can still be utilized to guide the search to the next optimal solution, and last winning node can possibly serve as a good starting point for the next search. Accordingly, after each t_{update}, the $simulated$ attribute of each node is reset so that when a previously simulated node is selected, SINR-EM will be performed again for the current users' distribution and the node's score is updated accordingly. The $visits$ count is also reset for all nodes. Furthermore, consider that the previously winning node is given by $node_{\text{win}}$, then after simulating this node and the $root$ node again, the achieved scores are compared to select the node with the highest score as the current root to search from. This is called best root selection (BRS) (see Algorithm 4).

4.4 Simulations and discussion

To test TP-MCTS performance, two different simulation scenarios with different settings and baseline methods are considered:

1. **Scenario 1:** Through extensive simulations, the results achieved by TP-MCTS are compared to the results of a popular PSO-based solution mentioned in [80] for uniformly distributed users.

2. **Scenario 2:** Using the RWP [32], TP-MCTS is simulated against the baseline settings in which DBSs are randomly distributed. On the other hand, TP-MCTS updates the location of drones every t_{update} seconds in accordance with subsection 4.3.2.

Algorithm 3 Transmission Power Monte Carlo Tree Search.

Input: $\mathcal{U}$, M, $P_{\max}$, P_{Δ},

Output: $\boldsymbol{P}_T$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, .., \boldsymbol{\mu}_M]$

$root \leftarrow \text{NODE}([P_{\max}, .., P_{\max}], 0)$
$S_{max} \leftarrow 0$
$\boldsymbol{P}_{\boldsymbol{best}} \leftarrow [P_{\max}, .., P_{\max}]$
for $iteration < iteration_{max}$ **do**
    $root$.SIMULATE$(0, \mathcal{U})$
end for
object NODE$(\boldsymbol{P}_T, S_{parent})$
    $visits \leftarrow 0$
    $simulated \leftarrow false$
    $childs \leftarrow \{\}$
    $S \leftarrow S_{parent}$
    function SIMULATE$(S_{ancestor}, \mathcal{U})$
        if $simulated = false$ **then**
            perform SINR-EM (see Algorithm 1).
            S = obtain node score according to (4.1).
            if $S > S_{max}$ **then**
                $S_{max} \leftarrow S$
                $\boldsymbol{P}_{best} \leftarrow \boldsymbol{P}_T$
            end if
            for $i \leftarrow 1, M$ **do**
                $\boldsymbol{P}_{T_{child}} \leftarrow \boldsymbol{P}_T$
                $P_{T_{child},i} \leftarrow P_{T_{child},i} - P_{\Delta}$
                $childs \leftarrow \{childs, \text{NODE}(P_{T_{child}}, S)\}$
            end for
            $simulated \leftarrow true$
            if $S > S_{ancestor}$ **then return** S ▷ *win* occurred
            else if $S < S_{ancestor}$ **then return** 0 ▷ *loss* occurred
            end if
        end if
        $visits \leftarrow visits + 1$
        $child \leftarrow$ child with highest UCB value obtained according to (4.3)
        $result \leftarrow child$.SIMULATE$(\max\{S, S_{parent}\}, \mathcal{U})$
        if $result > 0$ **then** update score according to (4.2) ▷ update if *win*
        end if
        return $result$
    end function
end object

Algorithm 4 TP-MCTS with best root selection.

Input: $\mathcal{U}$, M, $P_{\max}$, P_{Δ},

Output: $\boldsymbol{P}_T$, $\boldsymbol{\mu} = [\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, .., \boldsymbol{\mu}_M]$

Perform TP-MCTS as defined in Algorithm 3.
At every $t = Nt_{\text{update}}$ s.t. $N \in \{1, 2, 3..\}$
    Set $visits \leftarrow 0$, $simulated \leftarrow false$ for all nodes.
    $node_{\text{win}} \leftarrow$ the node with highest score so far.
    $root$.SIMULATE$(0, \mathcal{U})$
    $node_{\text{win}}$.SIMULATE$(0, \mathcal{U})$
    if $node_{\text{win}}.score > root.score$ **then**
        for $iteration < iteration_{\max}$ **do**
            $node_{\text{win}}$.SIMULATE$(0, \mathcal{U})$
        end for
    else
        for $iteration < iteration_{max}$ **do**
            $root$.SIMULATE$(0, \mathcal{U})$
        end for
    end if

In both cases, the author discusses the results and deduces insights into the trade-off between different deployment parameters such as transmission power settings, users' quality of service, number of DBSs, and the achieved energy efficiency. Also, in both cases, given the required rate, R, for users and the expected capacity of DBS, $C_{\text{BS}} = B_D \times \gamma_{\text{SE}}$ where γ_{SE} is the assumed system average spectral efficiency, the maximum number of users per DBS can be obtained as

$$U_{\max} = \left\lfloor \frac{C_{\text{BS}}}{R} \right\rfloor. \tag{4.4}$$

4.4.1 Scenario 1 (PSO)

Description

In PSO, a number of particles is generated, whereas each particle entails a randomly generated set of parameters. In the case of the proposed DBS allocation algorithm in [80], each particle specifies a set of randomly generated 3D locations of DBSs. The particles converge with random speeds towards the most promising global (across all particles) and local (per individual particle history) solutions that minimize a given utility function. The chosen solutions are updated whenever a better one is found. Through the use of three different

utility functions, the problem of 3D allocation of DBSs is tackled by minimizing the first utility function until a specified value is achieved, and then minimizing the second utility function and so on until the third utility function achieves a value less than a given threshold. Initially, the number of required drones and drone capacity are estimated based on the assumption of a given system spectral efficiency, similarly to what is described above. Then, the three utility functions optimize for three objectives, successively; the first objective aims to preserve the capacity constraint of each DBS given the density of users in each sub-area. The second objective aims to achieve the required link reliability while preserving the first objective. Finally, the third utility function tries to improve the system's average spectral efficiency until it reaches the value used in the aforementioned assumption while preserving the first two objectives, and the algorithm finishes. At each iteration i and after finding the optimal global solution for the current objective (i.e., chosen from all positions that all particles tried), and the optimal local solution for each particle (i.e., the position that minimized the utility function for the individual particle), the velocity of each particle is obtained as

$$V^{i+1} = \phi^{\text{PSO}} V^{(}i) + c_1 \phi_1^{\text{PSO}} + (W_{\text{local}}^{(}i) - W^{(}i)) + c_2 \phi_2^{\text{PSO}} + (W_{\text{global}}^{(}i) - W^{(}i)), \tag{4.5}$$

where $V^{(}i)$ is velocity from the previous iteration, $W_{\text{global}}^{(}i)$ and $W_{\text{local}}^{(}i)$ are the global and local solutions respectively, and $W^{(}i)$ is the current particle position. Furthermore, c_1, c_2, ϕ_1^{PSO}, ϕ_2^{PSO}, and ϕ^{PSO} are the PSO arbitrary parameters.

Simulation Settings (PSO)

In this work, the results of the aforementioned PSO solution are compared to the proposed TP-MCTS algorithm assuming an area of 1 km $\times$ 1 km with 500 users. Simulation is done using different numbers of serving DBSs, beginning with the minimum number estimated by dividing the total number of users by the maximum number of users per DBS obtained in (4.4). Also, the SINR requirement of Γ_{th} is varied. The maximum number of iterations is limited to $I = 200$. Table 4.1 lists the simulation parameters. The PSO-related parameters are listed in Table 4.1(c) using the same notations as in [80]. ϕ (also called the inertia weight) is set as a linearly decreasing weight (LDW) [81] obtained as $\phi = \phi_{\max} - \frac{\phi_{\max} - \phi_{\min}}{I} i$, where i is the current iteration count, and ϕ_1 and ϕ_2 are randomly drawn uniformly from the specified ranges.

Table 4.1 (a) System parameters, (b) TP-MCTS parameters, (C) PSO parameters.

(a)

Parameter	Value
Number of users U	500
Path loss parameters α, β, η_{LOS}, and η_{NLOS}	9.61, 0.16, 1 dB, and 20 dB
Carrier frequency f_c and drone bandwidth B_D	2 GHz and 20 MHz
Noise power σ_0^2, required rate R, and spectral efficiency γ_{SE}	-110 dBm, 1 Mbps, and 2 bps/Hz
Link reliability Φ and SINR threshold $\Gamma_{\text{th}}^{\text{dB}}$	0.95 and -2/ 0 dB

(b)

Parameter	Value
Softmax scale $\varkappa$ and dropout	4 and 0.5
SINR difference S_{th}	
Drone height h and exploration constant K_e	70 and 0.01
Power step P_Δ and maximum power $P_{\max}$	100 mW and 500 mW

(c)

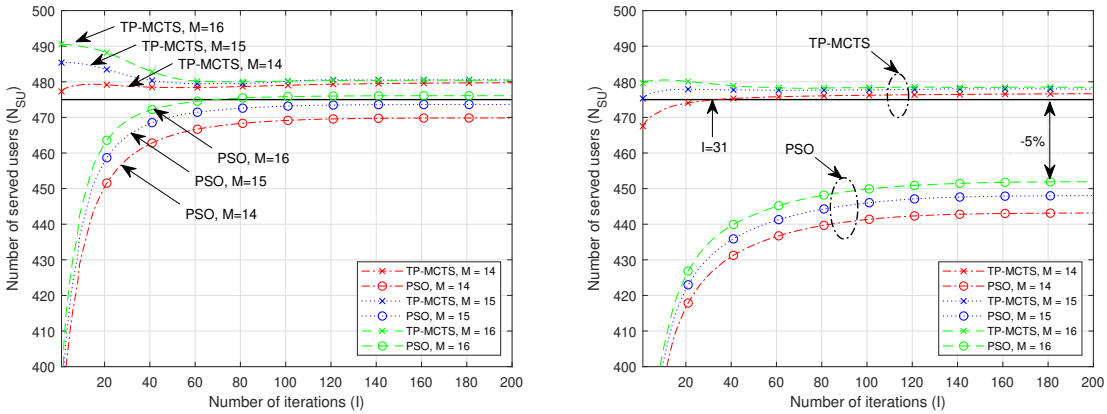
Parameter	Value
$\phi_{\min}$, $\phi_{\max}$, c_1, and c_2	0.9, 0.95, 0.095, and 0.15
ϕ_1 and ϕ_2	$\in$ [0.8 1.8]

Number of Served Users (N)

Figure 4.2 shows the value of N per iteration achieved by each algorithm for different numbers of serving DBSs, M, and two different conditions of Γ_{th}. In Figure 4.2a, it can be observed that for $M = 14$ and $M = 15$ the PSO solution was, on average, not able to provide the required $N_{\text{th}} = 475$ (indicated by the straight line). On the other hand, it appears that clustering users using SINR-EM without power control did already satisfy the requirement, and TP-MCTS changes the transmission power of DBSs in such a way that N is rather decreased while optimizing for the achieved energy efficiency ζ (see (4.1)). Additionally, Figure 4.2b plots the results given $\Gamma_{\text{th}}^{\text{dB}} = 0$ dB. Evidently, the PSO algorithm was far from finding a solution that satisfies the requirements, whereas TP-MCTS was able to find a solution for $M = 14$, and SINR-EM had already satisfied the requirements in the cases of $M = 15$ and $M = 16$.

Energy Efficiency (ζ)

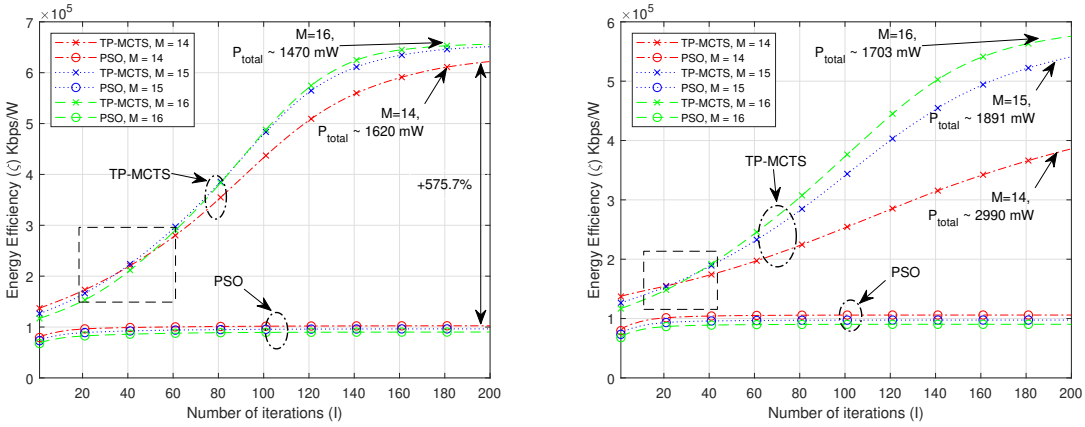
For $\Gamma_{\text{th}}^{\text{dB}} = -2$ dB, Figure 4.3a shows that TP-MCTS is able to substantially improve the energy efficiency by around 575%. Furthermore, increasing the number of DBSs from $M = 14$ to $M = 16$ decreases the total required transmission power, which is expected when cells are more dense. We observe that a higher number of DBSs initially results in degraded



(a) N per iteration for $\Gamma_{\text{th}}^{\text{dB}} = -2$ dB, $N_{\text{th}} = 475$ (solid line). (b) N per iteration for $\Gamma_{\text{th}}^{\text{dB}} = 0$ dB, $N_{\text{th}} = 475$ (solid line).

Figure 4.2 TP-MCTS vs. PSO by N. TP-MCTS always achieves the required Γ_{th} and substantially outperforms the PSO solution.

system performance in terms of ζ but later improves further than the cases with less DBSs as TP-MCTS optimizes the transmission powers.



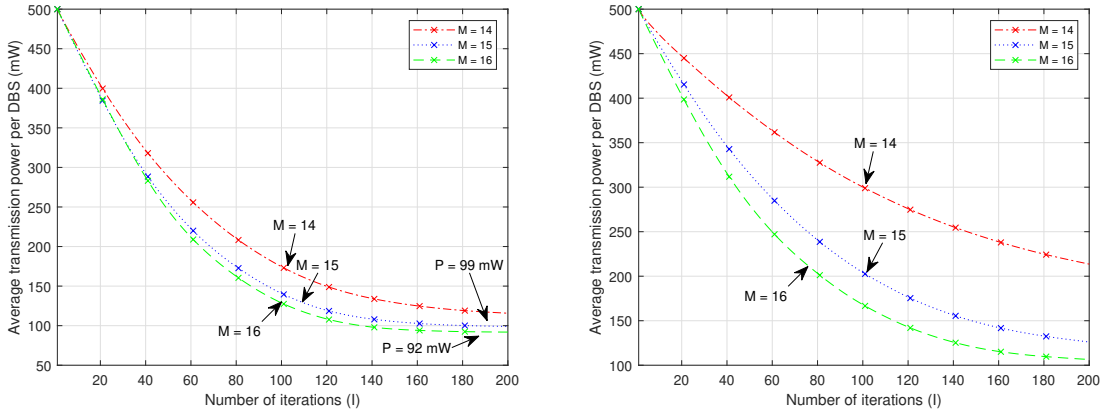
(a) ζ per iteration for $\Gamma_{\text{th}}^{\text{dB}} = -2$ dB. (b) ζ per iteration for $\Gamma_{\text{th}}^{\text{dB}} = 0$ dB.

Figure 4.3 TP-MCTS vs. PSO by ζ.

Transmission Power (P_T)

The average transmission power for all DBSs per iteration is plotted in Figure 4.4. For the case when $\Gamma_{\text{th}}^{\text{dB}} = -2$ dB, TP-MCTS seem to substantially decrease the transmission power of all DBS to almost 100 mW. This improves the interference levels between DBSs as can be deduced from previous plots. By examining the results when $M = 16$, we observe that

after a substantial number of iterations the average power per DBS is below 100 mW. Since $P_\Delta = 100$ mW (see Table 4.1) this indicates that TP-MCTS results in completely turning off a DBS by setting its transmission power to zero.



(a) Average P_T for $\Gamma_{\text{th}}^{\text{dB}} = -2$ dB.

(b) Average P_T for $\Gamma_{\text{th}}^{\text{dB}} = 0$ dB.

Figure 4.4 Average transmission power per DBS. Increasing the number of DBSs allows less transmission power per DBS.

4.4.2 Scenario 2 (RWP)

Simulation Settings

The author simulated the proposed TP-MCTS solution for a moving set of users generated according to the RWP model [82] against the baseline scenario where DBSs are distributed randomly. 200 moving users were considered in an area of 0.5×0.5 Km^2 that are served using 5 or 6 DBSs. TP-MCTS is run every $t_{\text{update}} = 5$ seconds for 20 iterations (see 4.3.2). 1000 trials are run, and the duration of each simulation trial is 120 seconds (i.e., 120 seconds of network functioning in real life). Table 4.2 lists the simulation parameters.

Number of Served Users (N)

Figure 4.5a shows the achieved N for 5 and 6 serving DBSs. The random approach achieves a constant performance on average, whereas TP-MCTS performance seems to decay between update intervals, and increases up to a maximum value after every update. TP-MCTS seems to also preserve a steady performance, which means that the tree is indeed able to adapt to the new users' locations and find new solutions. Figure 4.5b shows the average achieved ζ for 5 and 6 serving DBSs along with using the random distribution approach versus TP-MCTS. Clearly, when adding an extra DBS, the achieved energy efficiency decreases, but

Table 4.2 (a) System parameters, (b) TP-MCTS parameters.

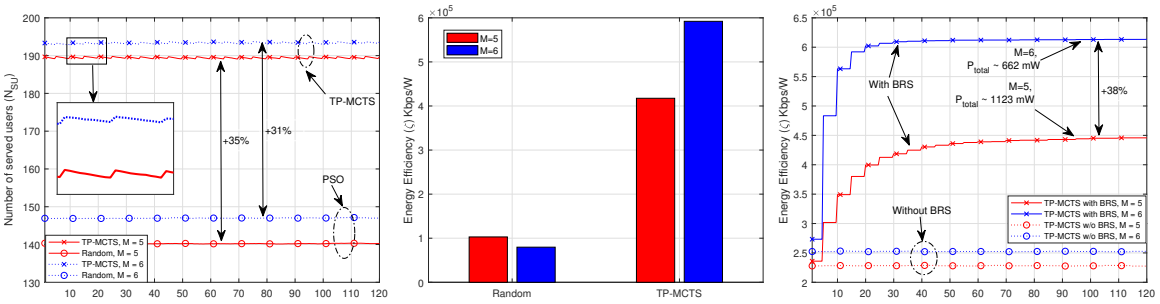
(a)

Parameter	Value
Number of users U	200
Path loss parameters α, β, η_{LOS}, and η_{NLOS}	9.61, 0.16, 1 dB, and 20 dB
Carrier frequency f_c and drone bandwidth B_D	2 GHz and 20 MHz
Noise power σ_0^2, required rate R, and spectral efficiency γ_{SE}	-110 dBm, 1 Mbps, and 2 bps/Hz
Link reliability Φ and SINR threshold $\Gamma_{\text{th}}^{\text{dB}}$	0.95 and 0 dB

(b)

Parameter	Value
Softmax scale $\varkappa$ and dropout SINR difference S_{th}	4 and 0.5
Drone height h and exploration constant K_e	70 and 0.01
Power step P_Δ and maximum power $P_{\max}$	100 mW and 500 mW
t_{update} and number of iterations	5 s and 20

in TP-MCTS it can be intelligently deployed to improve the performance of the system. Figure 4.5c plots ζ per second obtained using TP-MCTS with and without best root selection. The improvement resulting from using BRS is evident. The lack of such improvement in the case of no BRS is expected for a low iterations number of 20 since the tree search would never reach deep states that sufficiently reduce the transmission powers of DBSs.



(a) N per second. (b) Average ζ. (c) ζ per second.

Figure 4.5 Results for the second simulation scenario with time updates.

Collision avoidance

The author also measured the minimum distance that occurred between any DBSs during all trials. Recall that setting S_{th} to a small value reduces the overlap between clusters in SINR-EM and thus increases the distance between DBSs (see Section 3.2.5). In the case of $M = 5$,

the minimum distance recorded was 119.5 m, whereas for $M = 6$ the minimum distance was 107 m (see Figure 4.6). An interesting remark in the context of collision between DBSs is that TP-MCTS, if run indefinitely, will always avoid the state where two DBSs locations are equal, whereas turning one off completely (i.e., reducing power to zero) would always give a higher score (see (4.1)).

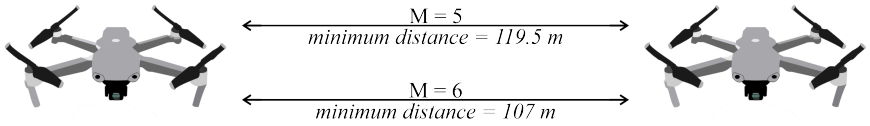


Figure 4.6 Minimum distance between any two DBSs during simulation.

4.5 Conclusion

It was shown that TP-MCTS outperformed the other baseline methods in terms of link reliability and energy efficiency. It was also shown that optimizing the transmission power yielded higher performance metrics in comparison with the PSO approach, which optimizes height. TP-MCTS is flexible and can optimize for whatever goals we set by defining the score appropriately. For example, the score given by (4.1) can be modified to include the system spectral efficiency instead of ζ, depending on the design requirements. Also, simulations showed that TP-MCTS can remove redundant DBSs by setting their transmission power to zero. Furthermore, the results proved that intelligent densification of cells can improve the system performance if properly managed. Finally, the author showed that TP-MCTS can utilize its tree history to guide the search towards more promising nodes.

Chapter 5

Backhaul-Aware DBS Placement

Chapter 3 addressed the problem of DBS placement by focusing solely on fronthaul-layer optimization aspects, such as the locations of GNs, path loss performance, and related parameters. In contrast, this chapter examines DBS placement from the perspective of backhaul optimization. The author considers a scenario in which deployed DBSs must ensure interconnectivity, either among themselves or with other network cells, via backhaul links, thereby maintaining overall backhaul connectivity. This formulation is particularly relevant in practical deployments where a group of UAVs operates as DRSs, providing backhaul links between two endpoints (e.g., a base station and a temporary cell established at a hotspot location).

What sets this backhaul interconnectivity problem apart from prior works on relay placement is the mobility and altitude flexibility of UAVs, which allow DRSs to establish LOS links with one another. Unlike terrestrial relays or fixed cells, DRSs are not restricted to predefined deployment locations, enabling more adaptable and efficient network topologies. This enables the use of directive channel technologies such as mmWave and FSO, which offer higher throughput, lower power consumption, and enhanced security. Another novel aspect that should be considered is the aerodynamic energy consumption of UAVs, which will limit the deployment time of DRSs, and which can be mitigated by energy harvesting technologies such as solar panels. This chapter addresses these novel aspects of relay deployment by introducing a framework for determining the optimal locations of DRSs employing directive transmission, while accounting for constraints related to the required backhaul data rate, drone energy consumption and harvesting through solar panels, and LOS clearance in urban environments with buildings acting as obstacles.

Some of the contributions in this chapter were published in:

1. Janji, S. & Sroka, P. *RIS-aided Multi-hop Backhauling for 5G/6G UAV-assisted Access Points*. Journal of Telecommunications and Information Technology, vol. 2, pp. 63–69, Jun. 2023. https://jtit.pl/jtit/article/view/1190

2. Janji, S. & Sroka, P. *Wykorzystanie Rekonfigurowalnych Inteligentnych Matryc Antenowych w Łączu Dosyłowym Sieci 5G/6G Wykorzystującej Bezzałogowe Statki Powietrzne*. Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, vol. 1, pp. 87–90, Aug. 2023. https://sigma-not.pl/publikacja-144743-2023-4.html

3. Janji, S., Samorzewski, A., and Kliks, A. *Procedura Rozmieszczenia Mobilnych Przekaźników Dosyłowych FSO Zasilanych z OZE*. Przegląd Telekomunikacyjny – Wiadomości Telekomunikacyjne, vol. 1, pp. 290–293, Aug. 2022. https://sigma-not.pl/publikacja-138994-2022-4.html

4. Janji, S., Samorzewski, A., Wasilewska, M. & Kliks, A. *On the Placement and Sustainability of Drone FSO Backhaul Relays*. IEEE Wireless Communications Letters, vol. 11, no. 8, pp. 1723–1727, 2022.

5.1 Related Work

Emerging cellular networks increasingly rely on small cells (SCs) as part of a densification strategy aimed at meeting high data rate and low latency demands [83]. However, SCs require robust backhaul connectivity to function effectively [84], and ensuring backhaul for mobile SCs remains a significant technical challenge. In this context, FSO communication links have demonstrated superior performance compared to RF links, offering higher data rates, more compact and lightweight antenna designs, and improved security [85].

Recently, attention has turned to deploying FSO transceivers on UAVs. In [43], the authors propose a statistical channel model for FSO communication between a ground transceiver and one mounted on a UAV. In [86], multirotor-based air-to-air (A2A) FSO communication is studied, showing that despite alignment challenges, commercial drones can maintain reliable links. The work in [87] investigates a network of DBSs equipped with FSO transceivers, where backhaul links are established via A2A FSO. Their results show notable throughput improvements when FSO is used for both fronthaul and backhaul.

Similarly, [88] approximates the ergodic rate for an FSO link between a UAV relay and a ground station, combining it with an RF-based ergodic sum rate to derive the relay's total capacity. In [89], UAVs serve as buffer-enabled FSO relays between static nodes, enhancing outage performance. However, these works typically assume clear LOS paths at sufficiently

high altitudes and do not address the placement challenges posed by urban obstacles. In contrast, this work considers environments where obstacles are explicitly mapped and may obstruct LOS paths, either due to building height or regulatory constraints that restrict UAV flight altitudes.

DRSs may also need to fly at low altitudes to minimize interference when establishing RF links with ground nodes, as suggested in [68]. Furthermore, existing works rarely account for the energy limitations of UAVs.

Flight time is inherently constrained by onboard battery capacity. To improve energy efficiency, some solutions optimize UAV trajectories, such as the work in [90], where a model for propulsion energy is derived and used for trajectory optimization. Other approaches explore renewable energy harvesting: [22] presents a UAV prototype equipped with solar panels, while [24] addresses SNR outage minimization by jointly optimizing transmit power and flight duration. In [47], trajectory and resource allocation are jointly optimized in the presence of cloud cover, with UAVs adjusting altitude to maximize solar energy capture. An energy-aware algorithm is proposed to adapt UAV flight based on aerodynamic consumption, available energy, and users' QOS demands.

However, this work considers that buildings may block both solar radiation and FSO signals. As a result, DRSs should be placed to avoid both types of obstruction.

The problem of positioning solar-powered DRSs to maintain backhaul connectivity across obstructed environments is addressed. First, Lee's visibility graph algorithm [91] is employed to determine feasible LOS paths between successive DRSs. Then, the algorithm is modified to prioritize sunny locations over shadowed ones. The results show that, in the given scenario, DRSs can be positioned in sunlit areas throughout daylight hours. By incorporating an energy model, the performance gains achieved by using solar panels are demonstrated. These are measured in terms of the reduced number of required recharging trips to maintain continuous backhaul connectivity.

5.2 Problem Formulation

A dense urban environment is considered with a MBS located at $L_M = (x_M, y_M)$ and a designated hotspot at $L_H = (x_H, y_H)$, as shown in Figure 5.1. Hotspot locations may be derived from user positions estimated via UE localization methods such as observed time difference of arrival [92] or GPS data. Alternatively, historical patterns and deep learning models may be used to predict them, as demonstrated in [93].

To enhance spectral efficiency and overall throughput, a dedicated DBS is deployed at the hotspot, with its backhaul connection to the MBS established via a set of solar-powered

DRSs using FSO links. Each link in the multi-hop chain must maintain unobstructed LOS. Figure 5.1 illustrates a scenario in which the link between the MBS and the first DRS is obstructed, while the link to the hotspot remains clear.

The case where DRSs are equipped with PV panels is also considered, thus generating energy when exposed to sunlight. Energy consumption and harvesting are modeled using the equations presented in Chapter 2: (2.29) for propulsion energy and (2.32) for solar power production.

The objective is to find a communication path from L_M to L_H that avoids all obstacles in the set of buildings illustrated in Figure 5.1. Given that the shortest unobstructed path in an environment with obstacles consists of straight line segments connecting the source to the destination via a (possibly empty) sequence of obstacle vertices [94], the problem can be formulated as a *visibility graph* problem [91]. In this formulation, the input graph G_o represents the set of geometric primitives (e.g., convex polygons defined by their vertices and connecting edges), and the output is a *visibility graph* G_v, which includes all edges connecting pairs of vertices in G_o that are *unobstructed*—that is, pairs between which a direct, obstacle-free LOS path exists. The final solution is obtained by applying a shortest path algorithm (e.g., Dijkstra's algorithm) to G_v, thereby identifying the optimal locations where DRSs should be deployed to maintain backhaul connectivity. In this work, Lee's algorithm is employed as described in [91] due to its conceptual simplicity and computational efficiency.

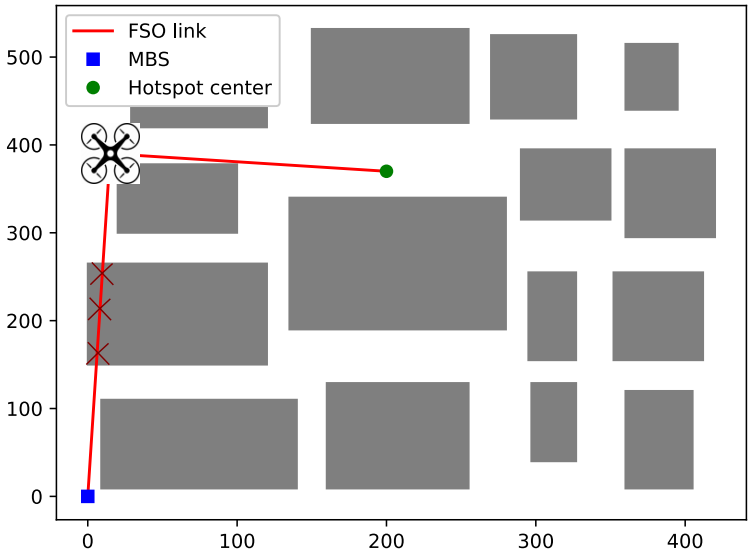


Figure 5.1 Illustration of the proposed scenario: high-rise buildings with MBS and hotspot destination. FSO links shuld have clear LOS.

5.3 Proposed DRS Placement Algorithm Without Solar Awareness

Given a map of the obstacles, a graph $G_o(V_o, E_o)$ is extracted, which contains all the vertices V_o and edges E_o that define the buildings (i.e., building corners and building edges in the top view, respectively). The two points defining the start and end locations given by L_M and L_H, respectively, are added to the set of vertices, V_o, so that their visibility tree is also computed. Then, Lee's algorithm is applied for each $v_i \in V_o$ to obtain the complete visibility graph G_v as follows [95]:

1. Initialize a horizontal scan line, $\vec{s}$, starting from the center vertex $c = v_i$ and pointing rightwards.

2. Sort all other vertices $\{v_j \in V_o \setminus v_i\}$ in ascending order according to the angle θ_j between $\vec{s}$ and $\vec{cv}_j$ (i.e., by rotating $\vec{s}$ counter-clockwise) and store them in a data structure A.

3. In another data structure, E_s, sort all the edges that intersect $\vec{s}$ in ascending order according to their distance from c, along with the distance magnitude. Clearly, the first edge is the only one that c can see.

4. Rotate $\vec{s}$ to every vertex v_j in the order of θ_j in A; at every vertex, check each edge that v_j belongs to, and either add or remove this edge from E_s depending on whether $\vec{s}$ would intersect it at the next rotation step.

5. At every rotation step, check whether the distance to v_j is less than the distance to the closest edge in E_s; if so, then v_j is visible and it is added to the visibility tree of v_i, otherwise it is not.

After obtaining $G_v(V_v, E_v)$, each edge $E_k = \{v_i, v_j\}$ is assigned a cost given by

$$D = \frac{d(v_i, v_j)}{d_{\max}} + 1, \tag{5.1}$$

where $d(v_i, v_j)$ is the Euclidean distance between the two endpoints, and $d_{\max}$ is a normalizing factor that limits the distance cost to the interval $[0, 1]$ and can be set to the maximum distance between any two points in the considered area. The cost is defined this way to prioritize minimizing the number of hops (i.e., number of DRSs), with distance acting as a secondary optimization criterion. The added value of 1 ensures that each chosen edge contributes to the hop count. Note that to ensure sufficient FSO rate, we merely need to

ensure that no edge is selected with a distance larger than $d_{\max}$, which is obtained based on a certain rate requirement using 2.26. A rate-aware approach is used in Section 5.5 below. Finally, Dijkstra's algorithm is applied on G_v using the edge costs defined in (5.1) to find the shortest communication path.

5.4 Proposed Solar-Powered DRSs Placement

To improve energy efficiency, the author of this thesis considers solar-powered DRSs and addresses the problem of their placement in a sunny environment with shadows cast by buildings. The objective is to leverage the energy harvesting capability to prolong the deployment duration of DRSs and reduce the number of trips required by each DRS to return to its base for battery recharging.

5.4.1 Sunny Points Search

Since the DRSs should be placed in sunny locations, the first step is to identify the set of candidate points that are exposed to sunlight at a given hour of the day (i.e., not shadowed by any building). A natural starting point that guides the search for the shortest path is the set of building vertices V_o used in the previous section. Given the sun altitude and azimuth angles, α_{solar} and ϕ_{solar}, and the set of obstacles defined by $G_o(V_o, E_o)$, we can test whether a given point falls within any shadow cast by a building in the direction $-\phi_{\text{solar}}$, with a shadow length of $\frac{h_{\text{B}}-h}{\tan(\alpha_{\text{solar}})}$, where h_{B} and h are the heights of the building and the UAV, respectively.

To find the set of sunny points V_s, the following steps are performed for every vertex $v_i = (x_i, y_i) \in V_o$. Let $v'_i = (x'_i, y'_i)$ denote a point in the interior of the polygon to which v_i belongs, such that v'_i is equidistant from the two edges connected to v_i, with a distance of D_T (see Figure 5.2). Around each v_i, a grid of test points is generated which is denoted by $V_T = \{(x_j, y_j) \mid x_j = x'_i \pm nD_T,\ y_j = y'_i \pm nD_T,$ for $n = 1, 2, .., C_T,$ and $(x_j, y_j) \notin \{(x'_i, y'_i), (x'_i, y_i), (x_i, y'_i), (x_i, y_i)\}\}$. The shift is subtracted along the x-axis when $x_T < x'_i$, and added otherwise; the same logic applies along the y-axis. The parameters C_T and D_T define the number of test points and their spacing, respectively. An example is illustrated in Figure 5.2. The initial point is discarded if it falls within a building. Points that lie exactly on building edges are also excluded to account for small structural features such as balconies or signage that could obstruct sunlight. These test points are examined sequentially, beginning with those closest to the building. If a sunny point is found, it is added to V_s and the search terminates for that vertex.

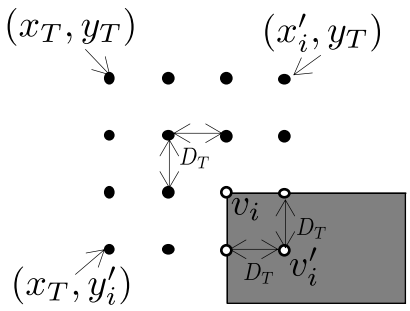


Figure 5.2 Test points grid for finding sunny spots for $C_T = 4$.

5.4.2 DRS Placement in Sunny Spots Algorithm

With the set of sunny locations V_s identified, Lee's algorithm, which was introduced in the previous section, is applied using an updated set of vertices defined as $V' = V_o \cup V_s \cup \{L_H, L_M\}$. This allows visibility trees to be computed for the newly added points in V_s. After obtaining the visibility graph, Dijkstra's algorithm is run with a modified cost function for each edge $E_k = \{v_i, v_j\}$ defined as:

$$D = \begin{cases} \frac{d(v_i,v_j)}{d_{\max}} + 1, & \text{if } v_j \in V_s, \\ \frac{d(v_i,v_j)}{d_{\max}} + 100, & \text{otherwise.} \end{cases} \tag{5.2}$$

The cost function in (5.2) penalizes the selection of shadowed spots by assigning them a significantly higher hop penalty (i.e., 100). This encourages the algorithm to favor sunny locations whenever possible.

5.4.3 Simulations and Discussion

The simulations considered a location in Madrid, Spain, to compute solar irradiation data over a 24-hour period. The environment model used is shown in Figure 5.1, where a backhaul link is provided to the drone serving users at the hotspot location L_H. The simulation parameters are listed in Table 5.1.

Placement of DRSs

Figure 5.3 shows the placement of DRSs along with the shadowed areas at six different times of the day, including the night when solar energy is unavailable. The algorithm successfully identifies paths consisting entirely of sunny points during daylight hours, avoiding any shadowed segments. A noteworthy observation is that between 7 am and 7 pm, only two hops are needed to cover a total distance of 617 m, whereas at night, three hops are required

Table 5.1 Simulation Parameters

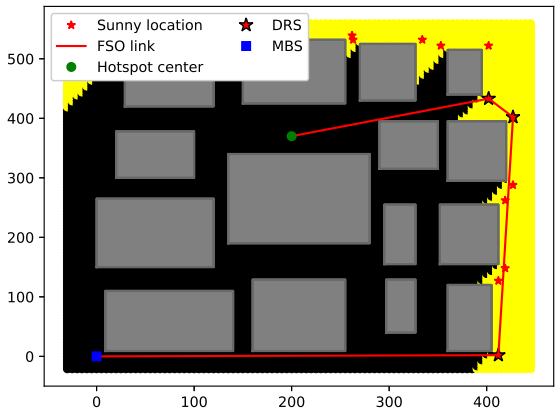
Parameter	Value
Longitude	3.70427144 °W
Latitude	40.41872533 °N
UAV hovering height (h)	20 m
Mean solar irradiation outside the atmosphere (I_0)	1353 W/m^2 [48]
Efficiency of solar cell (η_{PV})	0.2
Area covered by solar panels (S_{PV})	1 m^2 [47]
UAV mass (m)	4 kg [47]
Rotor disk area (A_D)	0.7854 m^2
UAV battery capacity	222 Wh [47]
FSOs backhaul power consumption (P_{FSO})	200 mW [87]
Distance cost normalization factor ($d_{\max}$)	700 m
Grid width for test points (C_T)	5
Grid spacing between test points (D_T)	7 m

to span a shorter distance of 490 m. This highlights that selecting obstacle vertices for pathfinding, while efficient in distance, may not always minimize the number of hops. Future work could explore alternative point selection strategies prioritizing hop count reduction and search efficiency, possibly leveraging precomputed sun maps.

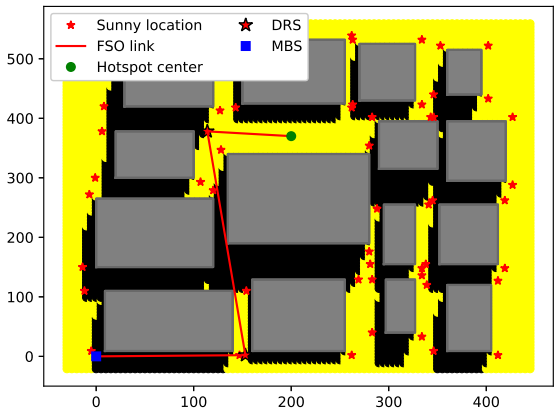
Sustainability

Figure 5.4a and Figure 5.4b show the number of new DRSs arrivals and returns from the deployment center per hour, respectively. Arrivals and returns can occur either for battery recharging or because a DRSs becomes newly required or redundant (e.g., note the extra arrival at 18:00 in Figure 5.4a corresponding to the additional DRSs in Figure 5.3e). Both figures compare cases with and without the use of renewable energy sources. It is evident that using solar panels significantly reduces the number of required trips during daylight hours when sunlight is available.

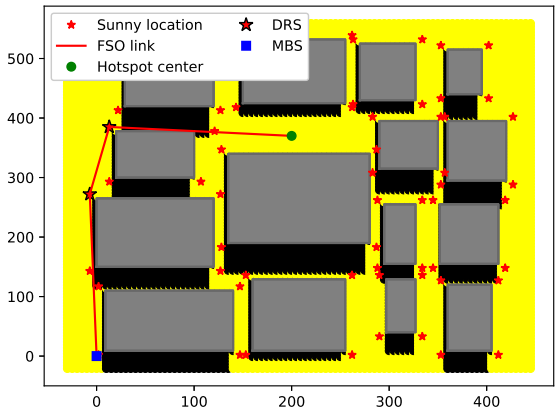
Over a full 24-hour period, 115 DRSs trips are required without PV panels, while only about 74 trips are needed with PV panels—a reduction of approximately 35%. This translates to a battery recharge energy savings of $(\frac{115}{2} - \frac{75}{2}) \times 222$ Wh = 4440 Wh. Additionally, depending on the distance between the deployment center and the DRSs, the energy saved on return flights can also be substantial in large-scale scenarios.



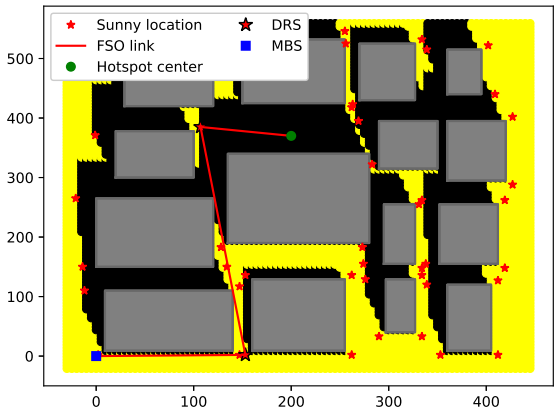
(a) 5 am, 3 DRSs hops



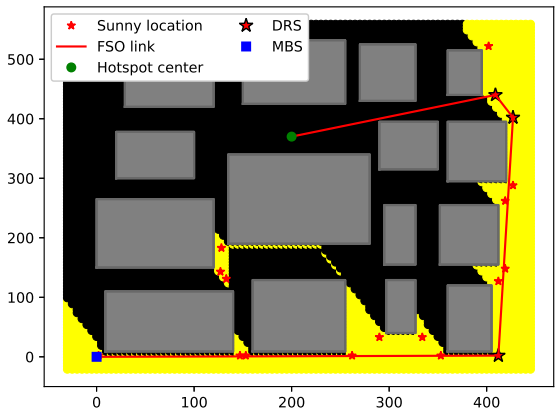
(b) 7 am, 2 DRSs hops



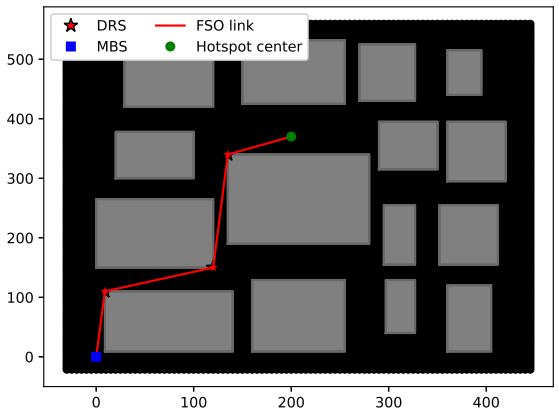
(c) 8 am, 2 DRSs hops



(d) 6 pm, 2 DRSs hops

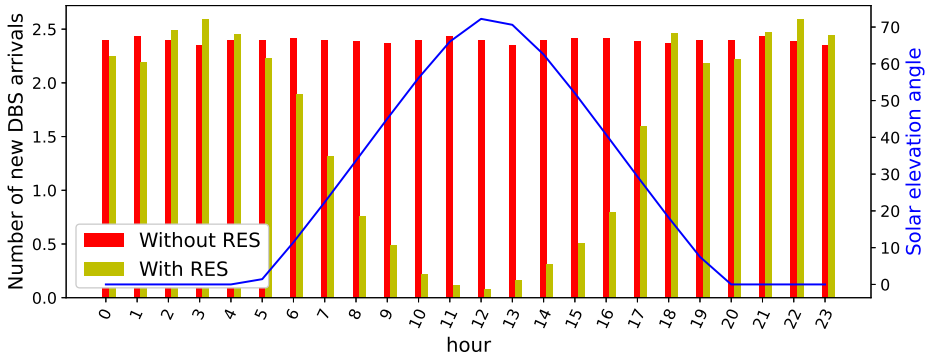


(e) 7 pm, 3 DRSs hops

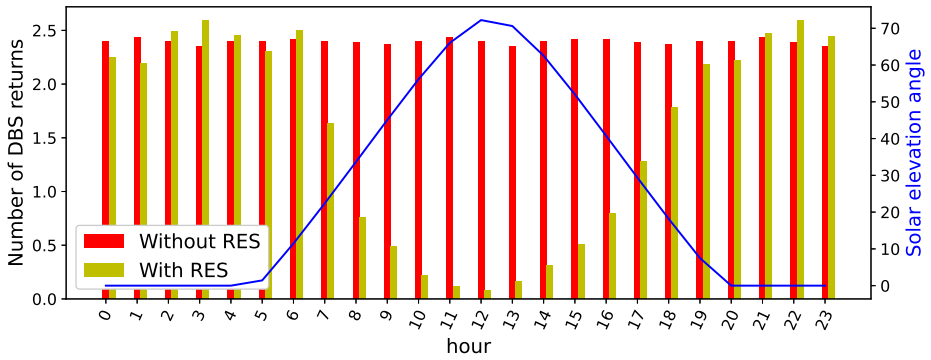


(f) Night, 3 DRSs hops

Figure 5.3 DRS placements at different hours of the day.



(a) Average number of new DRSs arrivals per hour.



(b) Average number of DRSs returns per hour.

Figure 5.4 Required number of DRSs trips to and from the charging station.

5.5 Proposed RIS-Aware DRS Placement

The role that RISs play in emerging networks and their advantages were highlighted in Section 2.4. In this section, their role in providing virtual LOS backhaul links between cells (including DRSs deployed in urban areas as explained above) is studied. The previously proposed algorithm is extended to account for the presence of RISs, which increases the connectivity achievable in an urban area by extending the visibility of endpoints as shown below.

In [96], the authors propose an energy-efficient aerial backhaul architecture that leverages a RIS mounted on a high-altitude platform. The system aims to establish reliable backhaul connectivity to UAVs-BSs operating in urban environments, particularly where terrestrial backhaul links may be obstructed. The study focuses on optimizing the placement of the aerial RIS, partitioning of the antenna array, and tuning the phase shifts of individual elements to maximize energy efficiency and ensure consistent connectivity. Similarly, [97] investigates a scenario in which multiple UAV-mounted RIS units are deployed to extend mmWave

coverage across a network of hotspots in environments prone to signal blockage. The authors formulate a centralized budget-constrained multi-player multi-armed bandit (BCMP-MAB) framework to govern the distribution and selection of hotspots by UAVs, aiming to maximize system throughput while conserving UAV battery life and avoiding coverage overlap. Both works avoid addressing urban obstacles by assuming sufficient operating altitude, and neither considers a hybrid architecture combining RIS and DRSs.

5.5.1 Problem Formulation

A slightly modified formulation is adopted here to investigate the practicality and advantages of deploying RISs for interconnecting relay cells. Specifically, it is assumed that at most M DRSs can be deployed to establish backhaul connectivity between two endpoints in an urban scenario modeled after Plac Ratajskiego in Poznań, Poland, using building layout data obtained from OSM [34]. An MBS is placed at the center of the area, as illustrated in Figure 5.5. The objective is to connect the MBS to various endpoints in the area (e.g., the hotspot represented by the green circle). For the purpose of analysis, in the simulation section, the vertices of buildings are treated as candidate hotspot locations to which backhaul paths should be established (see, for example, Figure 5.6 and Figure 5.7). These vertices are also assumed to be candidate locations for DRS deployment.

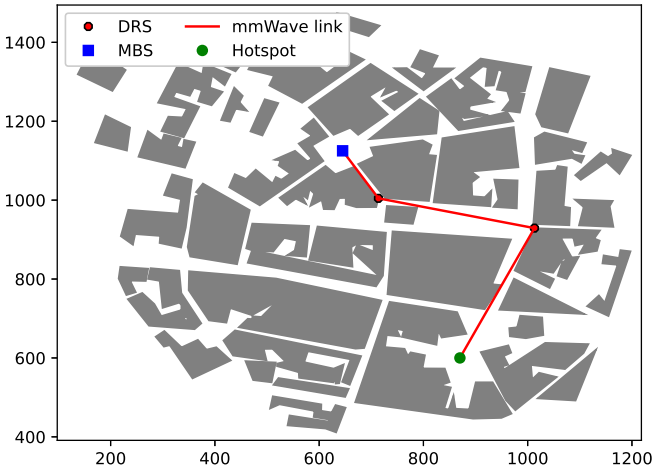


Figure 5.5 A building map of Plac Ratajskiego in Poznan, Poland, obtained from OSM [34], with an MBS placed at the center.

To analyze the impact of path loss more directly, the mmWave path loss model given by (2.20) is employed for the links between DRSs in this part of the study. This model results

in higher attenuation and lower throughput compared to FSO. For modeling the virtual links established via a RIS, the model presented in (2.28) is used.

In contrast to the previous section, the required throughput that each backhaul link must maintain is accounted for through defining the minimum constraint $C_{\min}$. The throughput of the i-th backhaul link is computed based on the corresponding path loss using the modified Shannon formula [98]:

$$C_i = \eta^B B^{\text{eff}} \log_2 \left(1 + \frac{P^{\text{TX}}}{L_i^{\text{linear}} \sigma^2} \right) \tag{5.3}$$

where η^B is the throughput efficiency of the system (i.e., the ratio of data bits to the total number of transmitted bits), B^{eff} is the effective bandwidth, P^{TX} is the transmission power of the backhaul transceivers, σ^2 is the noise power at the backhaul receivers, and L_i^{linear} is the path loss in linear scale—computed either from (2.20) for direct links or from (2.28) for RIS-assisted links. Thus, given the constraint $C_{\min}$, only links that satisfy

$$C_i \geq C_{\min} \tag{5.4}$$

are eligible to be used for establishing a backhaul connectivity path.

5.5.2 DRS Placement with RIS-Based Virtual LOS Links

To account for the presence of RISs installed in the area, the previously introduced algorithm is modified as follows. After constructing the visibility graph (i.e., using Lee's algorithm) for the set of vertices defined by the endpoints, building vertices, and, if applicable, sunny vertices for the solar-aware scenario, the edge cost function is updated before applying a shortest path algorithm. The edge cost, previously defined in (5.2), is now modified as:

$$D_{\text{edge}} = \begin{cases} L_{\text{direct}}^{\text{dB}} + P & \text{if direct edge and } C_{\text{edge}} > C_{\min} \\ L_{\text{RIS}}^{\text{dB}} + P & \text{if RIS edge and } C_{\text{edge}} > C_{\min} \\ \infty & \text{otherwise} \end{cases} \tag{5.5}$$

where $L_{\text{direct}}^{\text{dB}}$ is computed using (2.20), $L_{\text{RIS}}^{\text{dB}}$ is given by (2.28), P is a fixed hop penalty to promote paths with fewer relays, and ∞ is a sufficiently large value representing that the edge does not satisfy the minimum rate requirement and should therefore be avoided whenever possible.

5.5.3 Simulations and Discussion

In the simulations, two scenarios are compared: the case where there are no RISs, and the case where two RISs are installed in a square at marked locations, as will be shown below. These RISs provide virtual LOS paths between any two points that are visible to the RIS within the obstacles, the channel through the RIS also includes a beamforming gain as previously mentioned. These points belong to the set of feasible DRS locations obtained from the buildings' corners. Table 5.2 lists the simulation parameters assumed.

Table 5.2 Simulation parameters.

Parameter	Value
Transmission power P^{TX}	100 mW
RIS beamforming gain g_{bf}	20 dB
Noise power σ^2	−131 dBm
Throughput efficiency η	0.82
Effective bandwidth B^{eff}	18.72 MHz
Minimum throughput C_{min}	210 Mbps

The first set of results, shown in Figure 5.6, presents the required number of DRS hops to provide backhaul connectivity from the MBS to each point in the considered set. In this scenario, the author assumes that $C_{\text{min}} = 210$ Mbps and no RISs are installed. For comparison, Figure 5.7 shows the results when two RISs are placed at predefined positions within the area. The deployment of RISs clearly improves the reachability of the MBS to more distant endpoints and reduces the number of required hops to reach previously connected locations (compare with Figure 5.6).

Next, the achievable throughput is evaluated at each point in the map, and the results are compared across different scenarios. Figure 5.8 presents heatmaps that illustrate the distribution of achievable rates in the considered area. To compute the throughput at any point, the selected backhaul path is identified, and the minimum rate among all hops along that path is obtained as the maximum achievable rate at the destination.

Figure 5.8 demonstrates the performance improvement in terms of both reachability and throughput when only a single RIS is added to the system. These gains stem from creating new virtual LOS paths and the increased signal strength due to the beamforming gain of the RIS. Interestingly, while adding more RISs enhances the throughput in regions near the square, it can reduce the throughput in more distant areas on the right side of the map. This suggests that the introduction of new shorter paths with fewer hops can sometimes lead

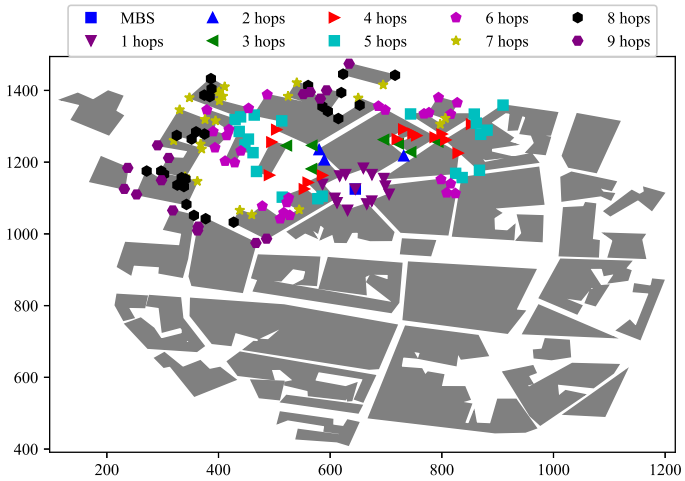


Figure 5.6 Required number of DRS hops to reach each point, indicated by the color and shape of the markers. Results correspond to the case with no RIS deployment.

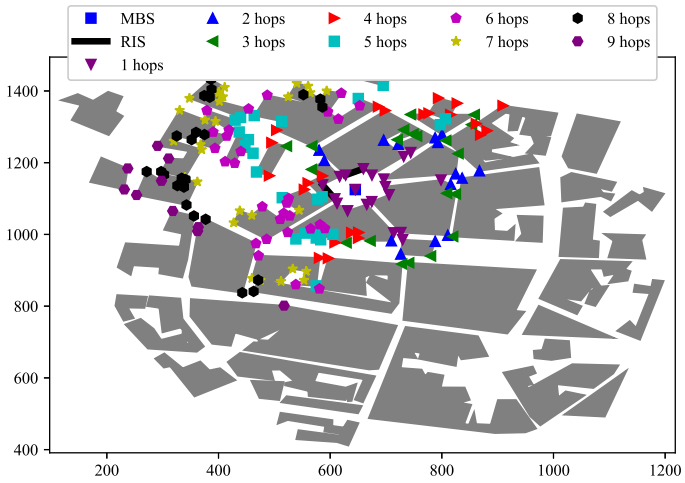


Figure 5.7 Required number of DRS hops to reach each point when two RISs are deployed at the marked locations.

to reduced end-to-end throughput if one of the new hops has significantly lower capacity compared to the previously optimal path.

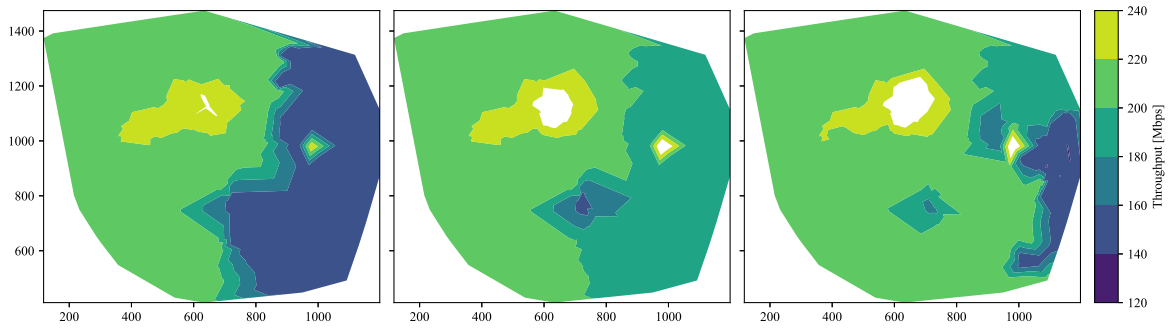


Figure 5.8 Heatmaps illustrating the achievable rate distribution across the map. Left: no RISs deployed. Middle: one RIS installed. Right: two RISs installed at the positions shown in Figure 5.7.

5.6 Conclusion

This chapter investigated the problem of backhaul-aware DRS placement in urban environments, focusing on the establishment of high-throughput, energy-efficient, and LOS backhaul connectivity. Unlike previous chapters, which concentrated solely on fronthaul optimization, here the author focused only on backhaul-related aspects, along with practical constraints related to energy consumption and environmental obstructions such as buildings.

The author introduced a visibility-graph-based framework for determining the optimal placement of DRSs under both basic and solar-powered configurations. Extending Lee's visibility algorithm enabled the incorporation of shadowed areas and the prioritization of sunny locations to improve energy sustainability. Simulation results demonstrated that this solar-aware placement strategy can significantly reduce the number of required recharging trips and extend UAV deployment time.

Furthermore, the use of RISs is integrated to augment the backhaul network with virtual LOS paths. This hybrid approach, combining DRSs and RISs, was shown to improve reachability and reduce the number of hops needed to connect distant nodes. The simulations also highlighted a trade-off between minimizing the number of hops and maintaining high throughput, depending on the placement and configuration of the RISs.

Chapter 6

DBS Placement with Joint Fronthaul and Backhaul Awareness

Chapter 3 addressed the problem of DBS placement from a fronthaul perspective, considering parameters such as GN locations, path loss performance, and interference from adjacent cells. Then, in Chapter 5, the focus shifted to the backhaul layer, where it was examined how to maintain connectivity between DRSs, a centralized MBS, and a hotspot location within urban environments. This was done while accounting for constraints related to energy consumption and harvesting, LOS clearance, and throughput requirements.

This chapter addresses both fronthaul and backhaul considerations jointly. Specifically, a framework for placing DBSs is proposed such that each node simultaneously ensures coverage for GNs under a minimum SNR constraint and satisfies a backhaul adjacency constraint. The latter requires that each DBS has at least N_{B} neighboring DBSs within backhaul communication range, enabling the formation of backhaul links for relaying user data. This constraint is commonly known as a k-neighbors constraint.

Satisfying the N_{B}-neighbors requirement guarantees that the resulting backhaul network is connected, which in turn enables the construction of valid backhaul routing paths. Increasing N_{B} not only enhances redundancy and reliability but also improves the flexibility of the formed backhaul mesh. Once placement is complete, technologies such as FSO can be employed to establish directed links between neighboring DBSs, enabling efficient routing of user data to designated endpoints. This class of problems has real-world applicability, with prominent examples including the *Loon* project [99], which used high-altitude balloons, and the *Free Basics* initiative by Meta [100], which aimed to extend connectivity to underserved regions using drones.

To solve the placement problem, HC algorithm is introduced based on a geometric interpretation of the dissimilarity matrix. This agglomerative HC procedure clusters the

GNs and determines the initial DBS placement, ensuring that each cluster center (i.e., DBS) maintains at least N_{B} neighbors. A larger number of neighbors improves the robustness of the backhaul network and leads to higher-quality solutions in the subsequent optimization step.

Finally, the backhaul mesh topology design is addressed in Chapter 7, where the problem of routing data across the selected set of DBSs is treated separately.

The contributions of this chapter have been submitted to the IEEE Internet of Things Journal and are available as a preprint in the public repository: Janji, S., Wawrzyniak, P., Formanowicz, P. & Kliks, A. *Integrating UAV-Enabled Base Stations in 3D Networks: QoS-Aware Joint Fronthaul and Backhaul Design*. arXiv preprint, 2024. https://arxiv.org/abs/2404.17547

6.1 Related Work

Extensive research has been conducted on DBS placement and backhaul provisioning, as surveyed in [1, 5, 10, 11]. Similar placement challenges arise in other domains such as sensor network coverage [101] and wireless mesh networkss (WMNs) [102], where access points must be positioned to provide user coverage while preserving multi-hop connectivity. The focus in this chapter is specifically on works addressing the placement of multiple wireless access nodes to ensure both user coverage and multi-hop backhaul connectivity. This problem is tackled using a novel HC algorithm.

6.1.1 Single-Hop and Two-Hop Backhaul Strategies

DBSs are typically modeled using coverage disks in $\mathbb{R}^2$, each defined by a radius within which GNs must fall to be considered served. Ensuring that every GN is covered by at least one DBS corresponds to solving the GSC problem [63] as previously described in Section 3.2.2. For example, [55] places DBSs incrementally along a spiral trajectory to cover all GNs, aiming to minimize the total number of DBSs.

However, coverage alone is not sufficient—DBSs must also maintain backhaul connectivity. In [103], DBSs connect directly to a MBS to meet backhaul demands, addressing both placement and resource allocation. Similarly, [104] considers a system in which DBSs are connected to a tethered balloon gateway, optimizing user associations, transmit powers, backhaul allocation, and placement to maximize end-to-end throughput.

The approach proposed below generalizes these models by allowing multi-hop backhaul, where DBSs forward each other's traffic. This introduces additional constraints, as DBSs

must be placed to cover GNs while maintaining connectivity for traffic relay. In [105] and [106], a single relay DBS assists others in forwarding traffic. In the former, placement and DBS-GN associations are alternately optimized, while the latter uses a GA for placement after random initialization. In contrast, the model considered in this chapter allows each DBS to serve GNs and simultaneously act as a relay.

6.1.2 Node Placement Under Connectivity and Coverage Constraints

Problems involving joint user coverage and node connectivity have also been studied in WMNs [102, 107–110]. In [107], the deployment area is discretized into a grid, where each grid point is a candidate location weighted by local GN density. Placement is initialized using fixed patterns, and an evolutionary algorithm refines the solution based on a fitness function that evaluates coverage and connectivity. Similarly, [108] employs a greedy heuristic to select grid points that maximize the same fitness function.

In [110], candidate locations are ranked by profit (e.g., population density), and constraints are imposed such that some nodes are mandatory and require two disjoint paths to a gateway. The objective is to maximize profit under a budget constraint on node count, solved via a three-stage heuristic involving initial selection, linear programming, and greedy graph expansion. These approaches, however, do not guarantee complete GN coverage. By contrast, the proposed HC algorithm ensures that every GN lies within the coverage area of at least one DBS.

Moreover, the HC method avoids grid-based restrictions. Instead, it clusters GNs that are near to each other and places DBSs at the resulting cluster centroids. Clusters merge hierarchically while satisfying both coverage and connectivity constraints, guided by a geometric interpretation of the dissimilarity matrix. For connectivity, a k-neighbor constraint is enforced: each DBS must have at least k neighboring DBSs or MBSs within backhaul range. To the best of the author's knowledge, this is the first use of HC to enforce both adjacency (connectivity) and range (coverage) constraints through geometric dissimilarity measures.

Similar placement problems also appear in sensor networks [101, 111, 112]. In [111], artificial potential fields guide nodes until each maintains exactly k neighbors, optimizing for area coverage rather than specific targets. However, this method has high computational complexity [101].

In [112], the objective is to ensure that each sensor covers a specific target and remains connected to a base station through a single neighbor. Connectivity is preserved by restricting node movement to maintain edges from a precomputed relative neighborhood graph. Although effective, this algorithm operates iteratively, adjusting node positions based on con-

nectivity checks, and is therefore computationally intensive and suitable only for small-scale problems. Additionally, it enforces only a 1-neighbor constraint ($k = 1$), whereas the below work supports general k-neighbor settings (defined by N_B).

6.2 System Model

At any time instance, GNs follows the PCP distribution model introduced in Section 2.1.1. Furthermore, there is a subset $\mathcal{M}$ of M deployed DBSs, selected from a larger set of available DBSs denoted by $\mathcal{A}$. The deployment is governed by the binary selection variable $a(k)$ defined as

$$a(k) = \begin{cases} 1 & \text{if DBS } k \text{ is deployed,} \\ 0 & \text{otherwise} \end{cases} \tag{6.1}$$

where $k \in \{1, 2, \ldots, A\}$, and $A = |\mathcal{A}|$ is the cardinality of the candidate set, assumed to be sufficiently large. The objective is to minimize the number of selected DBSs by minimizing $|\mathcal{M}| = M = \sum_{k=1}^{A} a(k)$. Additionally, as defined in Chapter 2, the set of all deployed DBSs and MBSs is given by (2.19).

To characterize the channel between a GN and its serving DBS, the model from Section 2.2.1 is adopted. Rather than maximizing the SINR, the goal is to ensure that each GN achieves a minimum SNR, corresponding to a maximum path loss threshold. Let $L_{\max}$ denote the maximum permissible PL, determined by the transmission power and a predefined SNR. As described in [38], this parameter enables the derivation of the maximum achievable coverage radius $R_{\max}$ at a given DBS height. Initially, the coverage radius of each DBS, denoted by $R_{\mathrm{A}}(j)$, is set to $R_{\mathrm{A}}^{0} \leq R_{\max}$, corresponding to an initial height h_0. As discussed before in Section 3.2.6, the height of each DBS can then be individually optimized to either reduce or extend $R_{\mathrm{A}}(j)$. A GN n is considered to be served by DBS j if it lies within the corresponding coverage disk. This is expressed by the binary variable

$$W'(j,n) = \begin{cases} 1 & \text{if } d_{2\mathrm{d}}(j,n) < R_{\mathrm{A}}(j) \wedge d_{2\mathrm{d}}(j,n) = \min\limits_{j'} d_{2\mathrm{d}}(j',n), \\ 0 & \text{otherwise} \end{cases} \tag{6.2}$$

where $d_{2\mathrm{d}}(j,n)$ is the 2D distance as defined in (2.2). This ensures that each GN is associated with at most one DBS, i.e., $\sum_{j=1}^{M} W'(j,n) \leq 1$ for all $n \in \{1, \ldots, U\}$. Note that shadowing, fading, and interference are not considered in this study. The focus is solely on coverage and backhaul throughput. For example, such effects can be handled through scheduling or spectrum reuse among clusters via lower-altitude DBSs.

6.3 Problem Formulation

Given the coverage disk radius of each DBS and a predefined minimum number of required backhaul neighbors N_{B}, the placement problem is defined as follows: minimize the number of deployed DBSs, ensure full coverage of all GNs, and satisfy the connectivity constraint that each DBS must be within a distance $d_{\max}$ of at least N_{B} other BSs. Increasing the value of N_{B} leads to denser backhaul connectivity, allowing greater flexibility for constructing feasible backhaul routes, which is addressed later in Chapter 7.

The following indicator function is defined to represent potential backhaul links

$$W^{\mathrm{FSO}}(i,l) = \begin{cases} 1 & \text{if } d(i,l) \leq d_{\max}, \\ 0 & \text{otherwise}, \end{cases} \tag{6.3}$$

where $d(i,l)$ is the Euclidean distance between BSs i and l.

The optimization problem is then formulated as

$$\min_{a(k),\, \mathbf{m}_j,\, b(i,l)} \sum_{k=1}^{|\mathcal{A}|} a(k) \tag{6.4a}$$

$$\text{s.t.} \quad \sum_{j=1}^{M} W'(j,n) = 1, \quad \forall n \in \{1,\ldots,U\}, \tag{6.4b}$$

$$\deg(v_j) = \sum_{\substack{i=1 \\ i \neq j}}^{M+B} W^{\mathrm{FSO}}(i,j) \geq N_{\mathrm{B}}, \quad \forall j \in \{1,\ldots,M+B\}, \tag{6.4c}$$

where $W'(j,n)$ is the binary GN-to-DBS mapping variable defined in (6.2). The objective function (6.4a) seeks to minimize the total number of deployed DBSs. The first constraint (6.4b) ensures that each GN is served by exactly one DBS, while the second constraint (6.4c) guarantees that every DBS has at least N_{B} neighboring BSs within the backhaul communication range $d_{\max}$.

Similarly to the reasoning in Section 3.2.2, if the backhaul range $d_{\max}$ is relaxed to infinity, the connectivity constraint becomes trivially satisfied, reducing the problem to a pure GSC formulation. The addition of the N_{B} neighbor constraint generalizes the classical GSC problem, and therefore, the overall problem remains NP-hard.

6.4 Hierarchical Clustering (HC) for Selecting Number of DBSs and Their Locations

As in Chapter 3, the assumption is that all DBSs are initially deployed at a common altitude h_0, which results in a uniform coverage radius R_{A}^0 for all DBSs (see Section 3.2.6). At a later stage, and once clustering is performed using the proposed HC algorithm, the altitude of each DBS can be individually adjusted to improve the SNR of its associated GNs, as previously detailed in Section 3.2.6.

Then, an agglomerative HC is utilized, which is a bottom-up method that starts with each data point as an independent cluster and progressively merges them based on similarity or distance metrics. This approach, with the proposed modifications, ensures the number of neighbors constraint, N_B, as specified in (6.4c), and guarantees that the clustered GNs fall within the coverage radius $R_{\mathrm{A}}(j)$ of the serving DBS. The unweighted pair group method using centroids for agglomerative hierarchical clustering (UPGMC) [113] method is adapted within the agglomerative HC framework to specify the quantity and positions of DBSs. These adaptations introduce additional merging criteria to meet the defined constraints. Beginning with the 2D location of each GN as a separate cluster, the algorithm then merges clusters based on Euclidean distance until a single cluster is formed or a predetermined stopping criterion is met. The clustering steps are registered in a linkage matrix, which saves the clustering history. Then, using this matrix, any number of clusters can be selected for the final results.

Firstly, a dissimilarity matrix of size $U \times U$ is initialized to capture the pairwise 2D distances among all GNs, using the formula in (2.2). This same distance computation is also applied between the centroids of any two clusters, C_a and $C_{a'}$, where $a, a' \in \{1, \ldots, U\}$. The symmetric dissimilarity matrix at iteration t, denoted $\boldsymbol{D}^t(\mathcal{U})$, is given by:

$$\boldsymbol{D}^0(\mathcal{U}) = \begin{bmatrix} 0 & d_{2\mathrm{d}}(1,2) & \cdots & d_{2\mathrm{d}}(1,U) \\ d_{2\mathrm{d}}(2,1) & 0 & \cdots & d_{2\mathrm{d}}(2,U) \\ \vdots & \vdots & \ddots & \vdots \\ d_{2\mathrm{d}}(U,1) & d_{2\mathrm{d}}(U,2) & \cdots & 0 \end{bmatrix}. \tag{6.5}$$

The entry $D_{a,b}$ in row a and column b refers to $d_{2\mathrm{d}}(a,b)$. At each iteration t, rather than searching the full matrix, the algorithm evaluates a subset $\mathcal{P}$ of candidate entries. The selection criteria for $\mathcal{P}$ are introduced later. The pair of clusters yielding the minimum value from this subset:

$$D_{a,b}^t = \min_{p,q \in \mathcal{P}} D_{p,q}^t \tag{6.6}$$

are merged into a new cluster. The distances between this newly formed cluster and all others are then recalculated, resulting in the updated dissimilarity matrix $\boldsymbol{D}^{t+1}(\mathcal{U})$. If $C_q = C_a \cup C_b$ is the newly formed cluster, the updated distance to another cluster C_s follows the UPGMC method [113]:

$$d_{2\mathrm{d}}(q,s) = \frac{n_a}{n_a+n_b} d_{2\mathrm{d}}(a,s) + \frac{n_b}{n_a+n_b} d_{2\mathrm{d}}(b,s) - \frac{n_a n_b}{(n_a+n_b)^2} d_{2\mathrm{d}}(a,b), \tag{6.7}$$

where n_a and n_b are the respective cardinalities of C_a and C_b. This UPGMC update ensures the dissimilarity matrix reflects the squared Euclidean distances between cluster centroids [113], which corresponds to DBS-to-DBS distances as used in (2.2).

In the modified UPGMC algorithm proposed by the author, a binary mask matrix $\hat{\boldsymbol{D}}^t$ is computed using a function $F\colon \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$, where:

$$\hat{D}^t_{a,b} = F(D_{a,b}) = \begin{cases} 1 & \text{if } d_{2\mathrm{d}}(a,b) \le d_{\max} \wedge a \neq b, \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

Summing each column of $\hat{\boldsymbol{D}}^t$ gives a vector containing the number of cluster neighbors within range:

$$\boldsymbol{C}^0 = \begin{bmatrix} \sum_{a'=1}^{U} \hat{D}^0_{1,a'} \\ \vdots \\ \sum_{a'=1}^{U} \hat{D}^0_{U,a'} \end{bmatrix}. \tag{6.9}$$

Assuming DBSs are placed at cluster centers at iteration t, and that the remaining number of clusters is $U-t$, then the following vector that yields the number of neighbors that each cluster has can be obtained as:

$$\boldsymbol{C}^t = \begin{bmatrix} \sum_{a'=1,a'\neq 1}^{U-t} \hat{D}^t_{1,a'} \\ \vdots \\ \sum_{a'=1,a'\neq U-t}^{U-t} \hat{D}^t_{U-t,a'} \end{bmatrix} = \begin{bmatrix} \sum_{a'=1,a'\neq 1}^{U-t} W^{\mathrm{FSO}}(1,a') \\ \vdots \\ \sum_{a'=1,a'\neq U-t}^{U-t} W^{\mathrm{FSO}}(U-t,a') \end{bmatrix}. \tag{6.10}$$

Using (6.8) and (6.10), we can make sure that no cluster has less than N_{B} neighbors as follows. Consider the following example: if C_c is a cluster with only one neighbor, C_a, (meaning $C_c = 1$) and $N_{\mathrm{B}} = 1$, any clustering of C_a may alter its central location. Merging it with a different cluster than C_c may relocate C_a out of C_c's vicinity, breaking the N_{B} condition. To avoid this, before merging clusters C_a and C_b, it should be ensured that every

non-zero element in the vector

$$\boldsymbol{V} = \left(\hat{D}^t_{:,a} \oplus \hat{D}^t_{:,b}\right) \circ \boldsymbol{C}^t, \tag{6.11}$$

is higher than N_{B}, where $\circ$ represents the Hadamard product and $\oplus$ signifies the binary OR operation. Thus, the following condition must be satisfied:

$$\forall V_i \in \boldsymbol{V},\ i \neq a, i \neq b : V_i > 0 \implies V_i > N_{\mathrm{B}}. \tag{6.12}$$

If not, the current pair is skipped, and the next pair satisfying (6.6) is considered. This modification ensures each resulting cluster maintains at least N_{B} neighbors.

To also guarantee that all GNs in a cluster fall within a coverage radius R^0_{A}, the complete linkage method for agglomerative hierarchical clustering (CL) HC update rule [113] can be leveraged. The distance from another cluster C_s to the merged cluster $C_q = C_a \cup C_b$ is

$$d_{2\mathrm{d}}(q,s) = \frac{1}{2}\left(d_{2\mathrm{d}}(a,s) + d_{2\mathrm{d}}(b,s) + |d_{2\mathrm{d}}(a,s) - d_{2\mathrm{d}}(b,s)|\right). \tag{6.13}$$

Applying this from $\boldsymbol{D}^0$ yields $\boldsymbol{R}^t$, where $R^t_{a,b}$ is the maximum distance between any two points from C_a and C_b [113].

Theorem 1. *If two clusters C_a and C_b are merged at iteration t such that*

$$R^t_{a,b} \leq R^0_A, \tag{6.14}$$

then all GNs in the new cluster are guaranteed to lie within a distance of R^0_A from the resulting DBS.

Proof. Let $C_c = C_a \cup C_b$ contain n_c elements, and let the furthest pair be $C_{c,A}$ and $C_{c,B}$, with squared distance

$$R^2_{\mathrm{A}} = (x_a - x_b)^2 + (y_a - y_b)^2. \tag{6.15}$$

Let the centroid be

$$\bar{P}_{C_c} = \left(\frac{1}{n_c}\sum_{i=1}^{n_c} x_i, \frac{1}{n_c}\sum_{i=1}^{n_c} y_i\right), \tag{6.16}$$

where (x_i, y_i) are the coordinates of point i in C_c. Then, the squared distance from any point P to the centroid is

$$d^2_{2\mathrm{d}}(P,\bar{P}) = \left(x_p - \frac{1}{n_c}\sum_{i=1}^{n_c} x_i\right)^2 + \left(y_p - \frac{1}{n_c}\sum_{i=1}^{n_c} y_i\right)^2. \tag{6.17}$$

Applying Jensen's inequality to the convex function $\Psi(x) = (C-x)^2$, where C is any real number, gives

$$d_{2\mathrm{d}}^2(P,\bar{P}) \leq \frac{1}{n_c}\sum_{i=1}^{n_c}\left((x_p-x_i)^2+(y_p-y_i)^2\right). \tag{6.18}$$

Since each term $(x_p-x_i)^2+(y_p-y_i)^2$ is bounded by R_{A}^2 from (6.15), then the inequality holds

$$d_{2\mathrm{d}}^2(P,\bar{P}) \leq R_{\mathrm{A}}^2. \tag{6.19}$$

□

Based on the theorem, merging any pair (C_a, C_b) must meet both (6.12) and (6.14). Therefore, the valid cluster merge candidates at iteration t form the subset

$$\mathcal{P}^t = \{(C_a, C_b) : (6.12) \text{ and } (6.14) \text{ hold}\}. \tag{6.20}$$

Algorithm 5 presents the step-by-step procedure of the proposed HC technique. In each iteration t, two clusters are selected and merged into a single cluster. A linkage matrix of dimensions $U \times 4$ is used to keep track of the clustering history. In this matrix, the first two entries of row t record the indices of the clusters that were merged during iteration t. The third entry stores the dissimilarity measure used to merge them, while the fourth entry indicates the total number of GNs contained in the resulting cluster. This linkage matrix is then used to reconstruct the hierarchical clustering structure.

6.4.1 Complexity of the Proposed Agglomerative HC Algorithm

At iteration t, there are $U-t$ clusters, resulting in up to $\binom{U-t}{2} = \frac{(U-t)(U-t-1)}{2}$ candidate pairs to evaluate. In the worst case, when $\mathcal{P}^t$ spans the entire matrix, the total number of comparisons throughout the clustering process is

$$\sum_{t=0}^{U-1}\binom{U-t}{2} = \sum_{n=1}^{U}\binom{n}{2} = \frac{(U-1)U(U+1)}{6}, \tag{6.21}$$

which leads to a worst-case computational complexity of $O(U^3)$. Since the computations of $\mathcal{P}^t$, $\boldsymbol{D}^t$, $\hat{\boldsymbol{D}}^t$, and $\boldsymbol{R}^t$ are each $O(U^2)$, they do not increase the overall asymptotic complexity.

However, an $O(U^3)$ algorithm may become intractable for large U. A commonly adopted workaround is *preclustering*, which reduces U by merging nearby GNs using lightweight methods such as grid-based spatial binning. This approach has $O(U)$ complexity and can drastically reduce the initial problem size without significantly losing spatial accuracy. Grid resolution can be tuned to balance between computational efficiency and placement fidelity.

Algorithm 5 DBSs Placement via Hierarchical Clustering

Input: GN locations $\boldsymbol{u}_n \; \forall n \in \{1, 2, \ldots, U\}$, backhaul range $d_{\max}$, minimum neighbor count N_B

Output: Number of deployed DBSs M, and their positions $\boldsymbol{m}_j \; \forall j \in \{1, 2, \ldots, M\}$

1: *Initialization*: Start with U clusters centered at the GNs. Compute:
   - $\boldsymbol{D}^0(\mathcal{U})$ using (6.5)
   - $\hat{\boldsymbol{D}}^0$ using (6.8)
   - $\boldsymbol{C}^0$ using (6.9)
   - $\mathcal{P}^0$ using (6.20)
2: Initialize $\boldsymbol{R}^0 \leftarrow \boldsymbol{D}^0$
3: **repeat**
4: Select clusters C_a and C_b from $\mathcal{P}^t$ that minimize (6.6)
5: Merge C_a and C_b
6: Update dissimilarities: compute $\boldsymbol{D}^{t+1}$ via (6.7), and $\boldsymbol{R}^{t+1}$ via (6.13)
7: Recompute $\hat{\boldsymbol{D}}^{t+1}$, $\boldsymbol{C}^{t+1}$, and $\mathcal{P}^{t+1}$ if necessary
8: **until** $\mathcal{P}^t = \emptyset$

Alternatively, lower-complexity clustering algorithms such as K-means or density-based spatial clustering of applications with noise data clustering algorithm (DBSCAN) (e.g., using KD-Trees) can also be employed for preclustering. These techniques generate a smaller set of input points for the hierarchical clustering process, improving runtime without compromising backhaul placement quality.

6.5 Simulation and Discussions

In the simulation below, a 10×10 km rectangular coverage area is considered, and for each simulation run, a new distribution of GNs based on the PCP model outlined in Section 2.1.1 is generated. In addition, the MBSs are located at the four corners of the area. The simulation parameters are listed in Table 6.1.

6.5.1 Number of Required DBSs, Number of Backhaul Neighbors, and GN-DBS Distance

To evaluate performance, the author simulated the proposed HC algorithm under varying values of N_{B} and the initial disk radius R_{A}. Figure 6.1 shows the number of DBSs required to meet the imposed constraints. This number corresponds to the resulting cluster count, $U - t$, once the subset $\mathcal{P}^t$ is exhausted at iteration t (see Algorithm 5). It is evident that for smaller

Table 6.1 Simulation Parameters

Parameter	**Value**
DBS height (h_0), MBS height, MBS count (B)	60 m, 25 m, 4
Environment variables (α, β)	9.61, 0.16 [38]
Expected PL (η_{LOS}, η_{NLOS}), carrier frequency (f_c)	1 dB, 20 dB [38], 2 Ghz, 4
PCP cluster standard deviation, PCP cluster density	100, 10^{-6}
Number of GNs per cluster	2

values of $d_{\max}$, the number of required DBSs increases non-linearly with N_{B}, reflecting the expected trade-off between higher backhaul redundancy and the total number of deployed nodes. However, as $d_{\max}$ increases, the sensitivity to N_{B} decreases, making R_{A} (i.e., the fronthaul coverage radius) the more dominant factor. This parameter can be controlled by adjusting the altitude h of the DBSs, as shown in Section 3.2.6.

In Figure 6.2, the performance of the HC clustering method is compared with K-means using K-means++ initialization [69]. The evaluation is conducted in terms of the number of backhaul neighbors per DBS and the distances between GNs and their associated DBSs. To isolate the effect of each constraint, the author set $R_{\text{A}} = \infty$ for neighbor count analysis and $d_{\max} = \infty$ for fronthaul distance analysis. While K-means is agnostic to such constraints, the proposed HC method inherently incorporates them. Both algorithms use the same number of DBSs, determined after convergence of the HC process.

The left panel of Figure 6.2 presents histograms of the number of backhaul neighbors for various N_{B} values with $d_{\max} = 1$ km. At $N_{\text{B}} = 2$, HC satisfies the constraint for nearly all clusters except for a few outliers with isolated GNs. Nevertheless, it consistently outperforms K-means in maintaining neighbor requirements. The right panel shows the distribution of GN-DBS distances. Although both methods yield similar distributions, K-means often produces associations that exceed the allowed coverage radius, while HC maintains compliance with R_{A}.

In summary, the proposed HC algorithm provides an effective and constraint-aware approach for DBS placement, balancing backhaul connectivity and fronthaul coverage without substantially increasing the number of deployed DBSs.

6.5.2 Height Optimization and Mobility Adaptation

As outlined in Section 3.2.6, the height of each DBS can be adjusted either to extend the coverage radius to accommodate GNs that move beyond the initial boundary, or to reduce the radius while maintaining coverage for all GNs, thereby minimizing overall path loss.

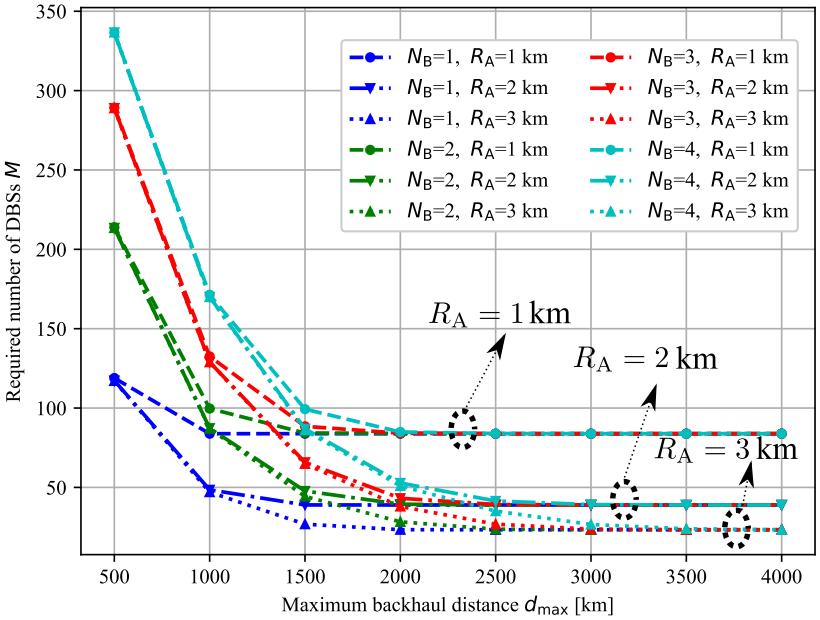


Figure 6.1 Average minimum number of required DBSs for various backhaul distances $d_{\max}$, under different N_B and R_A settings.

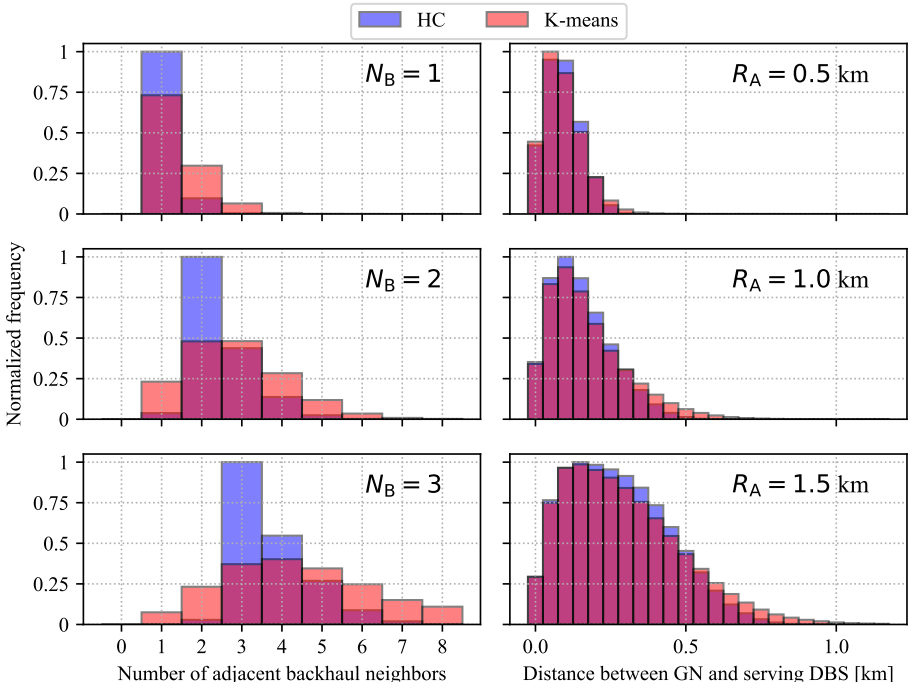


Figure 6.2 Left: Histogram of backhaul neighbors per DBS across various N_B values. Right: Histogram of GN-DBS distances under multiple R_A configurations. Comparison is made between the proposed HC and K-means, both using the same number of clusters.

Figure 6.3 illustrates the trade-off between height control, the number of required DBSs, and the surplus coverage radius, which is defined as $R_{\max} - R_{\mathrm{A}}(j)$. This quantity represents the distance between the farthest edge GN and the maximum achievable coverage radius. Figure 6.4 presents the improvement in SNR when the DBS height is optimized to reduce $R_{\mathrm{A}}(j)$. In this evaluation, $N_B = 1$, $d_{\max} = 3$ km, and $L_{\max} = 115$ dB, with a noise power of -147 dBm and a GN transmission power of 5 mW.

To generate Figure 6.3, different values of the initial height h_0, which determines the initial coverage radius R_{A}^0, along with the maximum allowable height $h_{\max}$, are used. As expected, smaller values of h_0 and R_{A}^0 result in a greater number of required DBSs, which increases in an exponential-like behavior. This also increases the surplus coverage margin, allowing more room for GN mobility. However, this increase seems to be near-linear. Similarly, increasing $h_{\max}$ expands the coverage margin and improves the system's ability to respond to GN mobility. Also illustrated is the importance of selecting the initial height parameter h_0 where a low value of $h_0 = 50$ m results in a substantially larger number of drones ($M = 190$) in comparison to the case when h_0 is increased beyond 140 m, where M decreases to around 60.

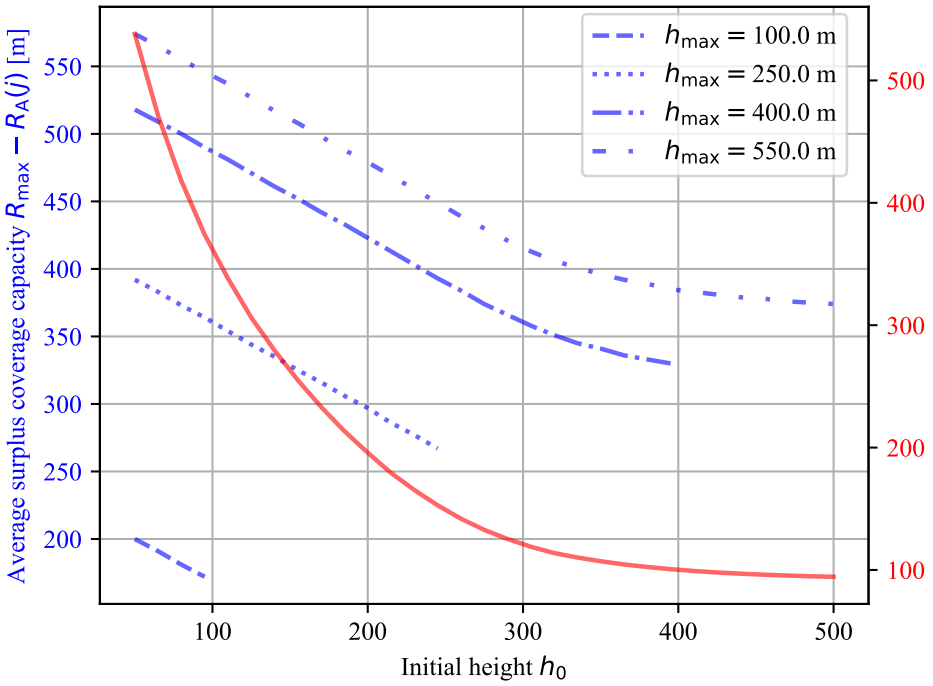


Figure 6.3 Left y-axis: average surplus coverage radius, defined as the mean distance between the farthest edge GN and the maximum coverage boundary. Right y-axis: required number of DBSs. Results are shown for different combinations of initial and maximum heights (h_0 and $h_{\max}$), and only for cases where $h_{\max} \geq h_0$.

Figure 6.4 presents the results of the proposed height optimization strategy under two different $h_{\max}$ constraints (50 m and 100 m). The plot shows the CDF of the SNR achieved by all GNs across multiple simulation runs of the HC algorithm. As expected, applying the height optimization technique improves the average SNR, as indicated by the vertical lines, and minimum SNR. Furthermore, a wider height range enables higher improvements.

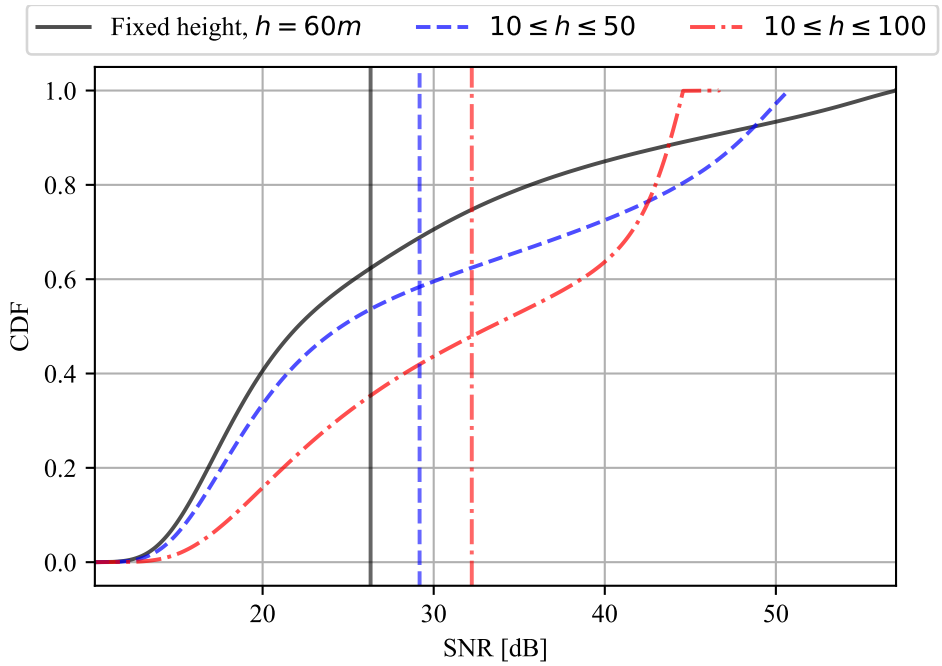


Figure 6.4 CDF of the achieved SNR at GNs, with and without height optimization. $R_{\mathrm{A}}^{0} = 1000$ km, $N_B = 1$, and $d_{\max} = 3$ km. The noise power is -147 dBm and transmission power to each GN is 5 mW. Curves are shown for a fixed height of 60 m (solid black), an optimizable height range of $[10, 50]$ m (dashed blue), and $[10, 100]$ m (dash-dot red). The mean values are also highlighted with vertical lines.

6.6 Conclusion

To solve this placement problem, the author of this thesis proposed a modified HC algorithm. This algorithm clusters GNs using a dissimilarity matrix and merges clusters hierarchically under two key conditions: the k-neighbors constraint and a disk radius constraint that guarantees each GN lies within the coverage range of its serving DBS. Simulation results confirm that the proposed method can meet both constraints without increasing the total number of required DBSs when compared to conventional approaches such as K-means. It was also demonstrated that initializing DBS heights allows us to manage the surplus coverage

radius, enabling dynamic adaptation to GN mobility through height adjustment. Simulation results revealed a tradeoff between surplus coverage and the number of required DBSs.

Furthermore, the asymptotic complexity of the proposed solution was analyzed and outlined methods to reduce computational burden through preclustering techniques. The output of this placement algorithm serves as an input for the subsequent backhaul mesh network design phase.

The next chapter builds on this placement outcome and addresses the problem of backhaul link selection and traffic routing, with the goal of ensuring end-to-end throughput guarantees across the DBS network.

Chapter 7

Designing the Backhaul Network - Final Joint Fronthaul and Backhaul Framework

Regardless of the deployment scenario or system parameters, DBSs serving GNs must establish backhaul connectivity to the core network. Unlike terrestrial base stations, DBSs rely on wireless backhaul links that may utilize FSO communication [87], cmWave or mmWave frequencies [114], or sub-6 GHz technologies. As outlined in [10], a DBS can connect to a terrestrial MBS or a satellite for backhaul. Additionally, DBSs can form direct wireless links with one another. A network consisting of interconnected DBSs, satellites, and MBSs forms what is now referred to as a three-dimensional (3D) network [27]. In such architectures, one or more MBSs or satellites that have direct access to the core network can serve as gateways, enabling other DBSs to relay their data through multi-hop paths.

Figure 2.1 illustrates these various configurations and backhaul strategies. The resulting topology, where nodes are interconnected through wireless links, forms what is known as a *wireless mesh network* [115] or a *wireless multi-hop network* [13]. These architectures are gaining relevance in modern communications, particularly due to their flexibility and scalability. In such topologies, cells may be self-organizing, enabling the network to dynamically reconfigure itself and reroute traffic to maintain connectivity in the face of failures or link disruptions.

Constructing reliable backhaul links is essential to avoid performance degradation due to bottlenecks in latency, throughput, or link reliability. In the context of airborne backhaul, FSO and mmWave technologies present promising options [44]. Depending on atmospheric

conditions and the operational altitude, FSO transceivers can sustain reliable links over distances exceeding 10 km.

However, designing backhaul mesh networks composed of DBSs introduces new challenges. Due to their elevated altitudes, DBSs operating in the same region often maintain LOS connectivity with one another, resulting in a dense graph of potential link pairs. The variability of point-to-point channel quality, driven by mobility and environmental sensitivity, complicates link reliability and topology stability. Furthermore, the mobility of DBSs may cause topology shifts, requiring the dynamic formation and termination of links. Unlike omnidirectional terrestrial base stations, establishing a backhaul link between two DBSs requires precise transceiver alignment. This alignment process depends on location data and is typically governed by a central controller, as is common in UAV and satellite networks [44]. Consequently, autonomous neighbor discovery—common in flying ad-hoc network (FANET) or internet of things (IOT) networks—is generally not feasible. In addition, the size, weight, and power constraints of UAVs limit the number of transceivers they can carry, thereby restricting the number of simultaneous backhaul connections per node.

This distinctive network model introduces a novel optimization problem: selecting a set of backhaul interconnections among a group of DBSs, subject to flow constraints and link limitations. Each DBS must be assigned N_B outgoing connections, and the cumulative downlink traffic generated by its associated GNs must be routed through the backhaul mesh to the gateway nodes, following the principle of flow conservation. Moreover, network robustness must be addressed by dimensioning the capacity margin available beyond the expected traffic load.

An illustrative example of this backhaul configuration problem is shown in Figure 7.1, where DBSs are placed across a region, and candidate backhaul links are indicated based on a predefined maximum link distance. Each link is annotated with the estimated throughput and aggregated load it must carry. The figure highlights two routing paths from different DBSs to a nearby MBS: one successful path (in green) that satisfies throughput demands, and one failed path (in red) where accumulated load exceeds the link capacity, resulting in congestion just two hops from the gateway.

In this chapter, DNP is formally defined as a MILP problem, where the objective is to identify a feasible backhaul configuration that satisfies both link degree and flow conservation constraints while minimizing congestion. After proving the problem's NP-hardness, a graph-based GA is introduced to solve it efficiently. The algorithm increases the redundancy of backhaul throughput and accommodates variable traffic distributions and link constraints. Simulation results confirm the effectiveness of combining the HC-based placement strategy

from the previous chapter with the proposed GA-based backhaul design, resulting in a comprehensive and scalable framework for joint fronthaul and backhaul optimization.

The contributions of this chapter have been submitted to the IEEE Internet of Things Journal and are available as a preprint in the public repository: Janji, S., Wawrzyniak, P., Formanowicz, P. & Kliks, A. *Integrating UAV-Enabled Base Stations in 3D Networks: QoS-Aware Joint Fronthaul and Backhaul Design*. arXiv preprint, 2024. https://arxiv.org/abs/2404.17547

7.1 Related Work

7.1.1 Backhaul Multi-Hop Mesh Topology Optimization

Following the placement of DBSs with method proposed in the previous chapter, the proposed GA algorithm is employed to determine the backhaul connections, ensuring that the cumulative throughput across the resulting multi-hop wireless mesh is successfully routed toward MBSs gateways, as illustrated in Figure 7.1.

In [116], DBSs are deployed to support small cells (SCs), forming a mesh backhaul network based on their initial positions. In this setting, the end-to-end backhaul capacity of a path is constrained by its weakest link. A game-theoretic approach is used in which each DBS autonomously selects its connections to optimize its backhaul throughput either with neighboring DBSs or directly with gateway MBSs. The DBSs locations are iteratively updated using a virtual force field model that incorporates attractive and repulsive forces. Results indicate that forming multi-hop paths achieves higher throughput compared to direct links.

In a similar study [117], the authors jointly optimize the placement, bandwidth, and power allocation of a fixed number of DBSs to construct a backhaul mesh with a single gateway node. The objective is to maximize the minimum throughput across all served GNs. The problem is tackled using alternating optimization and successive convex programming. However, neither [116] nor [117] guarantee minimum throughput values for individual GNs.

The work in [118] addresses a related scenario where UAVs provide connectivity to mobile robots in urban environments with obstacles, forming a tree-structured backhaul network rooted at an MBS. A game-theoretic algorithm is proposed to jointly optimize the backhaul topology (to reduce delay and improve throughput) and the UAV-robot associations. A virtual force field method is also used for collision avoidance and communication quality enhancement. However, the approach assumes a single MBS and does not guarantee end-to-

end connectivity to the core network. Furthermore, the number of backhaul connections per DBS is unconstrained in all three works [116–118].

In [119], the authors explore mmWave communications in FANETs and employ a generative adversarial network to optimize topology and select optimal communication paths while limiting each UAV to two connections. However, the framework does not consider the exact throughput requirements of GNs. A related study [120] focuses on backhaul topology design in a WMN with directional antennas. Routers and gateways are connected to maximize throughput satisfaction under load and distance constraints. A greedy tree-building algorithm is used, prioritizing the minimization of the network's maximum edge load. However, this simplified greedy approach does not consider the capacity of the selected edges, and minimizing the maximum edge load may not always be the optimal strategy. For instance, a high-load router with a high-capacity link to the gateway may be a better parent than a lower-load router with limited link capacity, yet the latter is chosen.

Similarly, [121] addresses DBS positioning to ensure full GN coverage while maintaining connectivity to terrestrial base stations, either directly or via other DBSs. The proposed method uses a low-complexity algorithm to select deployment sites from a fixed grid to maximize coverage. However, it does not incorporate throughput constraints and only considers the existence of a connection, not its quality or capacity. None of the aforementioned studies [119–121] consider the accumulation of relayed traffic across the mesh.

In contrast, the proposed GA ensures flow conservation: each forwarding link in the mesh must support both the local GN load and any relayed traffic from downstream DBSs, as visualized in Figure 7.1.

7.1.2 Conservation of Flow in Wireless Multi-Hop Backhaul

The study in [122] investigates tethered DBS backhaul using LOS mmWave links. Candidate deployment sites are predefined, and DBSs are allowed to relay traffic among themselves and to terrestrial stations while simultaneously serving GNs. A graph of feasible links is constructed, and for each GN, multiple shortest paths to the core are computed. Then, a deep reinforcement learning model selects backhaul routes to handle the aggregated traffic. However, the approach limits backhaul link selection to shortest paths, does not restrict the number of connections per DBS, and relies on a fixed set of deployment sites—limiting flexibility.

In [109], the authors present a formulation that resembles the considered problem. The objective is to deploy WMN access points and routers to cover all users and maintain backhaul flow to a core network. Routers and gateways can communicate if within range, and clients associate with access points within coverage. The system is modeled as a linear integer

program, solved using CPLEX, and aims to minimize deployment cost. However, like prior works, it does not limit the number of backhaul links per node and assumes a discrete set of candidate locations. Also, due to the problem's NP-hardness, exact backhaul optimization is not scalable for large instances.

In summary, this research addresses the backhaul design of a DBS-based network to serve GNs. Each DBS is limited to two full-duplex backhaul connections: one used for both relaying its local data and receiving forwarded load from upstream nodes, and a second for forwarding data to a downstream DBS. This design forms an acyclic wireless mesh, enabling unidirectional, path-based data flow from the core to every GN. The proposed approach guarantees flow conservation at each node and ensures that every link supports the cumulative data demands from its downstream connections, thus maintaining reliable and efficient network operation.

7.2 System Model

The model is an area in which MBSs and DBSs are deployed with determined locations. Each MBS can serve as a gateway to the core network, and all DBSs should have a connectivity path towards the core. For a given maximum backhaul distance constraint $d_{\max}$, the backhaul throughput between any two BSs i and l is modeled using the FSO rate expression $R_{\mathrm{FSO}}(i,l)$ given in (2.27).

7.2.1 Backhaul Network as a Graph

The backhaul mesh network is modeled as a graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ represents the set of all deployed BSs, and $\mathcal{E} = \{\{v_i, v_j\} \mid v_i, v_j \in \mathcal{V}, i \neq j\}$ denotes the set of all feasible wireless links between them. Each link is weighted by the achievable FSO throughput computed via (2.27). Each node j in the graph is tagged with the total traffic load from the GNs it serves, calculated using

$$G(j) = \sum_{n=1}^{U} W'(j,n) R_n, \tag{7.1}$$

where R_n is the fronthaul rate required by GN n.

Each DBS i forwards its aggregated traffic to a next-hop BS l, which can be either another DBS or an MBS, indicated by the binary variable:

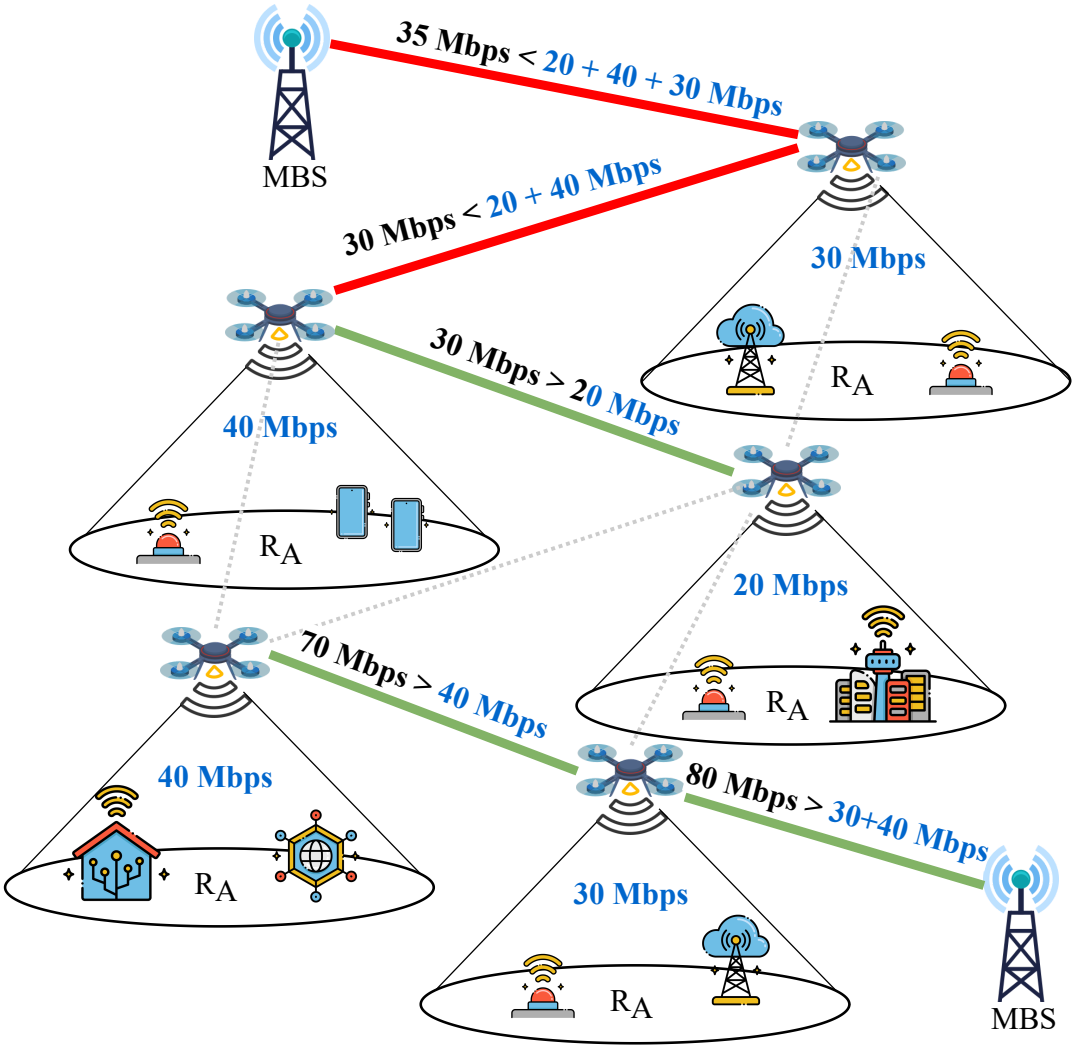


Figure 7.1 Visualization of the backhaul optimization scenario with fixed DBS positions and predefined GN traffic loads. Potential backhaul links are annotated with their corresponding throughput capacities and carried loads. Each DBS is labeled with the total traffic load from its associated cluster. Two example routing paths are highlighted: green links indicate sufficient backhaul capacity, while red links denote congestion, annotated with the respective cumulative traffic loads.

$$b(i,l) = \begin{cases} 1 & \text{if BS } l \text{ is the forwarding hop for DBS } i, \\ 0 & \text{otherwise} \end{cases} \tag{7.2}$$

Note that this relationship is directional; $b(i,l) = 1$ does not imply $b(l,i) = 1$.

The corresponding flow variable $f(i,l)$ represents the total traffic forwarded from DBS i to BS l, which includes:

1. The local traffic $G(i)$ from GNs directly served by DBS i,
2. Any relayed traffic received from other DBSs k that use i as their forwarding hop.

Hence, the total flow from i to l is:

$$f(i,l) = G(i) + \sum_{\substack{k=1 \\ k \neq i}}^{M} f(k,i). \tag{7.3}$$

Figure 7.1 illustrates an example of this backhaul mesh network with multiple DBSs, two MBSs, and interconnections limited by a maximum distance constraint $d_{\max}$ between any two backhaul neighbors.

7.3 Problem Formulation

Beyond simply fulfilling the current backhaul requirements, it is also important to provision surplus throughput along backhaul routes to accommodate future increases in GN traffic demands. This leads to the following problem formulation: Given a set of BSs, $\mathcal{V}$, with known positions; the achievable backhaul throughput between each pair of BSs as $R_{\text{FSO}}(i,l)$; and the load $G(j)$ associated with each GN cluster, the objective is to determine the backhaul mesh topology by setting the binary decision variables $b(i,l)$ introduced in (7.2). Considering the secondary objective of maximizing throughput redundancy, the optimization problem is defined as

$$\max_{b(i,l)} \quad \sum_{l=1}^{M}\left(\sum_{\substack{i'=1\\ i'\neq l}}^{M+B} R(l,i')b(l,i') - \sum_{\substack{i=1\\ i\neq l}}^{M} f(i,l)b(i,l) - G(l)\right) \tag{7.4a}$$

$$\text{s.t.} \quad \sum_{\substack{l=1\\ l\neq i}}^{M+B} b(i,l) = 1, \quad \forall i \in \{1,\ldots,M\}, \tag{7.4b}$$

$$\sum_{\substack{i=1\\ i\neq l}}^{M} b(i,l) \leq 1, \quad \forall l \in \{1,\ldots,M+B\}, \tag{7.4c}$$

$$\sum_{\substack{i=1\\ i\neq l}}^{M} f(i,l) + G(l) = \sum_{\substack{i'=1\\ i'\neq l}}^{M+B} f(l,i'), \quad \forall l \in \{1,\ldots,M\}, \tag{7.4d}$$

$$f(i,l) \leq b(i,l)R(i,l), \quad \forall i,l \in \{1,\ldots,M+B\}, i \neq l, \tag{7.4e}$$

$$\sum_{l=M+1}^{M+B}\sum_{\substack{i=1\\ i\neq l}}^{M} f(i,l) = \sum_{j=1}^{M} G(j), \tag{7.4f}$$

$$\sum_{l=1}^{M+B} b(i,l) = 0, \quad \forall i \in \{M+1,\ldots,M+B\}. \tag{7.4g}$$

Objective (7.4a) aims to maximize the surplus (i.e., redundant) backhaul throughput across the selected links. The constraints serve the following purposes:

- (7.4b): Ensures that each DBS is assigned exactly one forwarding hop.
- (7.4c): Guarantees that each BS can serve as a forwarding hop for at most one DBS, which, when combined with (7.4b), limits each DBS to a maximum of two backhaul connections and each MBS to only one.
- (7.4d): Enforces the flow conservation constraint, ensuring that the total data forwarded by a DBS equals the sum of its own GNs load and any relayed load from other DBSs.
- (7.4e): Ensures that each selected backhaul link has sufficient capacity to carry the total assigned load.
- (7.4f): Guarantees that the aggregate load from all DBSs is eventually delivered to the MBSs, confirming that all backhaul paths terminate at gateway nodes.
- (7.4g): Specifies that MBSs do not have forwarding hops, reflecting their role as end points in the backhaul paths.

Solving this optimization determines the backhaul structure by defining B directed sequences of BSs, represented as:

$$\mathcal{S} = \left\{\mathcal{S}^1, \ldots, \mathcal{S}^B\right\} \tag{7.5}$$

Each $\mathcal{S}^p$ denotes a multi-hop backhaul route comprising at least two BS indices:

$$\mathcal{S}^p = \left\{\mathcal{S}_1^p, \ldots, \mathcal{S}_{|\mathcal{S}^p|}^p\right\}, \tag{7.6}$$

where $\mathcal{S}_1^p \in \{1, \ldots, M\}$ is a DBS and $\mathcal{S}_{|\mathcal{S}^p|}^p \in \{M+1, \ldots, M+B\}$ is a MBS. Thus, each valid sequence begins with a DBS and ends at an MBS, forming a complete backhaul route to the core network.

This DNP can also be interpreted as a graph problem defined over the undirected graph $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{V} = \mathcal{B} \cup \mathcal{M}$ as introduced in Section 7.2.1. Each node $v_j \in \mathcal{M}$ has a corresponding load $G(j)$, and each edge $\{i, l\} \in \mathcal{E}$ is assigned a throughput weight $R_{\text{FSO}}(i, l)$.

The task is to identify a subgraph $G_S = (\mathcal{V}_S, \mathcal{E}_S)$ such that $\mathcal{V}_S = \mathcal{V}$ and $\mathcal{E}_S \subseteq \mathcal{E}$, while satisfying: - $\deg(v_i) \leq 2$ for all $v_i \in \mathcal{M}$, - $\deg(v_i) \leq 1$ for all $v_i \in \mathcal{B}$.

Moreover, there must exist a set of vertex sequences $\mathcal{S} = \{\mathcal{S}^1, \ldots, \mathcal{S}^B\}$ in G_S such that: - each node $v_i \in \mathcal{V}_S$ appears in at most one sequence, - each $v_i \in \mathcal{M}$ appears in exactly one sequence.

Furthermore, for every sequence $\mathcal{S}^p = (\mathcal{S}_1^p, \mathcal{S}_2^p, \ldots, \mathcal{S}_l^p)$ in $\mathcal{S}$, the following must be satisfied:

$$\mathcal{S}_1^p, \ldots, \mathcal{S}_{l-1}^p \in \mathcal{M}, \tag{7.7}$$

$$\mathcal{S}_l^p \in \mathcal{B}, \tag{7.8}$$

and

$$R(\mathcal{S}_i^p, \mathcal{S}_{i+1}^p) \geq \sum_{k=1}^{i} G(\mathcal{S}_k^p), \quad \forall i \in \{1, \ldots, l-1\}, \tag{7.9}$$

which corresponds to the flow constraint in (7.4e).

The DNP problem is computationally intractable, as established by the following theorem.

Theorem 2. *The DNP is NP-hard.*

Proof. To demonstrate the NP-hardness of the DNP, it is proven that its decision version is NP-complete. A problem is NP-complete if (i) its solution can be verified in polynomial time, and (ii) a known NP-complete problem can be transformed into it in polynomial time [123]. This is achieved by reducing the well-known Hamiltonian path problem (HPP) to an instance of the DNP.

In the HPP, one is given an undirected graph $G_h = (V_h, E_h)$ and must determine whether there exists a path $S_h = (v_1, v_2, \ldots, v_{|V_h|})$ that visits every vertex exactly once. To reduce this to the considered problem, a DNP instance is constructed with graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = V_h \cup \{b\}$, $\mathcal{M} = V_h$, $\mathcal{B} = \{b\}$, and $\mathcal{E} = E_h \cup \{\{v, b\} \mid v \in V_h\}$. Also, $G(j) = 1$ for all $v_j \in \mathcal{M}$ and define all link capacities $R(i,l) = |V_h|$.

If a solution to the HPP exists, then appending node b to the path yields a valid sequence $\mathcal{S}^h = (v_1, v_2, \ldots, v_{|V_h|}, b)$ in the DNP. Each link in this sequence supports the total accumulated load $G(i)$ without exceeding the capacity.

Conversely, any valid solution to the DNP must include a sequence ending in the gateway node b, since $\mathcal{B} = \{b\}$. This path, denoted $\mathcal{S}^b = (\mathcal{S}^b_1, \ldots, \mathcal{S}^b_{|V_h|}, b)$, must traverse all nodes in $\mathcal{M}$ exactly once. Removing b from this sequence yields a Hamiltonian path for G_h.

This reduction proves that the decision version of the DNP is NP-complete, and thus, the optimization version is NP-hard. □

Since the DNP is NP-hard, exact solutions are impractical for large instances. Therefore, the solution is to adopt an approximate approach using a GA to efficiently converge to near-optimal solutions.

7.4 Genetic Algorithm for Backhaul Optimization

A GA is considered, which is a class of metaheuristic algorithms well-suited for problems with vast search spaces or nonlinear characteristics that hinder traditional methods. Potential solutions are first encoded as chromosomes—data strings representing problem instances. Each chromosome is evaluated by a fitness function, which serves as a quantitative indicator of solution quality and guides the search toward promising regions of the solution space.

The GA employs three core operators inspired by natural genetic processes [124]:

1. *Selection*: Preferentially retains chromosomes with higher fitness, increasing the likelihood of propagating beneficial traits.

2. *Crossover*: Combines segments from pairs of chromosomes, enabling offspring to inherit features from both parents and fostering diversity.

3. *Mutation*: Introduces small random changes to chromosomes, enhancing diversity and aiding escape from local optima.

The following sections detail the implementation of the GA. This includes how solutions are encoded, the formulation of a custom fitness function tailored to the DNP, and the

calibration of genetic operators. Additionally, the termination criteria and heuristics used to enhance algorithm performance and robustness are outlined.

7.4.1 Representation

Each genome in the GA encodes concatenated backhaul paths as defined in earlier (similarly to (7.5)). Some sequences may consist of only a single MBS, indicating an unused node. Let DBSs be denoted by d_j for $j \in \{1, \ldots, M\}$ and MBSs by b_i for $i \in \{M+1, \ldots, M+B\}$. A sample genome may be $\mathcal{S} = \{b_7, d_6, d_1, d_4, d_5, b_8, d_2, d_3, b_9\}$, which corresponds to one unused MBS ($\mathcal{S}^1$), and two active backhaul sequences: $\mathcal{S}^2 = \{d_6, d_1, d_4, d_5, b_8\}$ and $\mathcal{S}^3 = \{d_2, d_3, b_9\}$. Genomes are ordered such that b_{M+B} appears last.

7.4.2 Initial Population Generation

The initial population is constructed by permuting DBSs and inserting MBSs randomly:

1. *DBS Permutations*: Generate $M!$ permutations of DBSs, representing all possible visitation orders.
2. *Random MBS Insertion*: Insert $B-1$ MBSs into each permutation at random positions between 0 and M, allowing duplicate insertions. The final MBS is fixed at the end.

This process yields $M! \times (M+1)^{B-1}$ total combinations. A subset of N_{initial} genomes is sampled to form the initial population.

7.4.3 Mutation

Mutation swaps two distinct random elements in the genome, preserving uniqueness while introducing diversity. Mutation frequency is tuned to balance exploration and solution stability.

7.4.4 Crossover

Crossover combines genetic material from two parent genomes by exchanging backhaul paths (ending at MBSs). To ensure validity, one parent is designated as the leader (whose sequences remain unchanged), and the other as the follower. Any duplicate elements from the follower are replaced by unused ones. Length discrepancies are resolved by adding or removing elements to preserve structural integrity.

7.4.5 Fitness

The fitness function steers evolution toward feasible, high-performing solutions. For each link in genome $\mathcal{S}$, residual capacity is computed as:

$$p'(\mathcal{S}_r^p, \mathcal{S}_{r+1}^p) = R(\mathcal{S}_r^p, \mathcal{S}_{r+1}^p) - \sum_{j=1}^{r} G(j), \tag{7.10}$$

where $\mathcal{S}_r^p$ is the r-th element in sequence p. A valid genome satisfies:

$$p'(\mathcal{S}_r^p, \mathcal{S}_{r+1}^p) \geq 0 \quad \forall r, p. \tag{7.11}$$

Two fitness functions are defined:

$$F_{\text{edge}}(\mathcal{S}) = \sum_{p=1}^{B} \sum_{r=1}^{|\mathcal{S}^p|-1} p'\left(\mathcal{S}_r^p, \mathcal{S}_{r+1}^p\right), \tag{7.12}$$

$$F_{\text{node}}(\mathcal{S}) = \sum_{p=1}^{B} \sum_{r=1}^{|\mathcal{S}^p|-1} \min_{k=1,\ldots,r} p'\left(\mathcal{S}_k^p, \mathcal{S}_{k+1}^p\right). \tag{7.13}$$

F_{node} ensures sufficient upstream capacity to support the load relayed through each node, whereas F_{edge} focuses on edge-wise residual capacity, favoring potentially strong crossover candidates.

7.4.6 Evaluation of Fitness Functions

Final solutions are evaluated using F_{node} with correctness enforced via (7.11). During evolution, genomes may be selected using either fitness function. Three different penalty mechanisms are tested for invalid solutions:

1. No penalty.
2. Subtract a fixed penalty from the fitness score.
3. Subtract the deficit in residual capacity (based on p').

This yields six total configurations:

1. Edge Surplus Without Penalty (ENP): fitness function based on edge surplus without penalty.

2. Edge Surplus with Value Penalty (EVP): fitness function based on edge surplus with a penalty value.
3. Edge Surplus with Edge Deficits Penalty (EEP): fitness function based on edge surplus with a penalty corresponding to deficits on edge flows.
4. Node Surplus Without Penalty (NNP): fitness function based on node surplus without penalty.
5. Node Surplus with Value Penalty (NVP): fitness function based on node surplus with a penalty value.
6. Node Surplus with Edge Deficits Penalty (NEP): fitness function based on node surplus with a penalty corresponding to deficits on edge flows.

7.4.7 Algorithm

The proposed GA is summarized by the pseudocode in Algorithm 6. It iteratively evolves a population of candidate backhaul configurations to identify the optimal or near-optimal solution. Starting with a randomly generated initial population of genomes, the algorithm evaluates each genome's quality using a defined fitness function. The fittest individuals are preferentially selected as parents for producing the next generation through crossover and mutation. To maintain diversity and avoid premature convergence, a small number of random mutations are introduced. Additionally, an elitism mechanism ensures that the best-performing individuals are preserved between generations. The process repeats until the stopping condition (e.g., maximum number of generations) is met. The best solution found is then returned.

7.4.8 GA Complexity

The complexity of the GA is determined by the number of generations, population size, and the cost of core operations. The fitness and mutation operations scale as $O(M+B)$, while crossover operations, which impose higher complexity because they require validation and potential fixing to ensure the validity of instances, are at $O((M+B)^2)$. The overall complexity is:

$$O\left(G \cdot \left(P(M+B) + P\chi_{\mathrm{M}}(M+B) + P\chi_{\mathrm{CO}}(M+B)^2 + P\chi_{\mathrm{E}}\right)\right), \tag{7.14}$$

Algorithm 6 Genetic Algorithm for Backhaul Optimization

Input: Population size N, stop condition SC, mutation rate χ_{M}, crossover rate χ_{CO}, fitness selection function F_{sel}, fitness solution function F_{sol}
Output: Optimized solution
1: Generate initial population of N_{initial} individuals.
2: **repeat**
3: Evaluate fitness for each individual.
4: Store best solution by F_{sol}.
5: Select parents based on F_{sel}.
6: Apply crossover with rate χ_{CO}.
7: Apply mutation with rate χ_{M}.
8: Apply elitism by preserving top χ_{E} individuals.
9: **until** SC
10: Return best solution.

where G is generations, P is population size, and χ_{M}, χ_{CO}, χ_{E} are mutation, crossover, and elitism rates respectively. Dominated by the crossover term, the complexity simplifies to $O((M+B)^2)$.

7.5 Joint Backhaul and Fronthaul Framework

The overall framework, illustrated in Figure 7.2, integrates the HC and GA algorithms, along with the height adaptation mechanism. For an initial height h_0, the HC algorithm determines the DBS placements with the minimum required number of DBSs, and then the GA computes the backhaul connectivity. If the GA fails, additional DBSs can be introduced by reversing merges in the linkage matrix, thereby expanding the backhaul solution space. After finding a feasible backhaul solution, the height of each DBS is optimized to reduce coverage radii and improve SNR. Afterwards, if (3.13) is violated due to GN mobility, the framework adapts in two ways: if the GN exits the current coverage radius but remains within the maximum possible radius $R_{\max}$, the DBS height is adjusted to extend coverage. Otherwise, HC is re-executed to recompute placement. Similarly, if (7.11) is violated due to load increase, GA is re-run to restore network flow conservation.

7.6 Simulations and Discussions

Following the simulation methodology described in Section 6.5, a rectangular coverage area of size 10x10 km is considered. For each simulation run, a new spatial distribution of GNs is

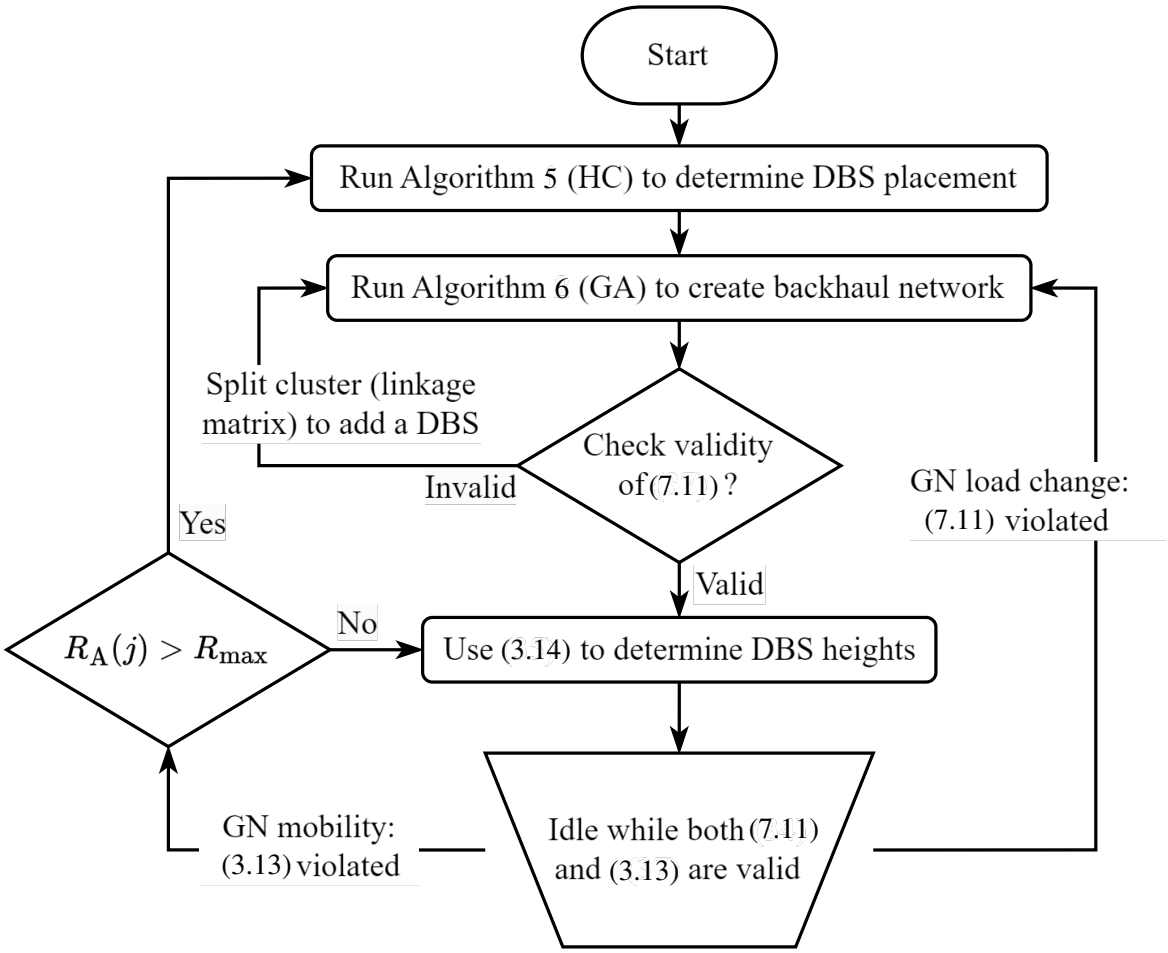


Figure 7.2 Flowchart illustrating the joint fronthaul and backhaul optimization framework.

generated using the PCP outlined in Section 2.1.1. The MBSs are statically positioned at the four corners of the area.

The simulations were implemented in Python. For solving the backhaul optimization problem, `PyGAD` library [125] is utilized to implement the proposed GA. To tune the GA parameters, preliminary tests on 20,000 randomly generated problem instances were conducted. Parameters were sampled from the following ranges:

- Maximum number of generations: 10–1000
- Initial population size: 10–1000
- Crossover rate (χ_{CO}): 1%–99%
- Mutation rate (χ_{M}): 1%–99%
- Elitism rate (χ_{E}): 1%–99%

- Parent selection method: random, rank, roulette wheel, steady-state, stochastic universal, tournament
- Stopping criterion (stagnant generations): 3–30

The final parameters were selected based on achieving a balance between convergence speed and solution quality. Although the chosen configuration performed well for the scenarios tested in this work, these parameters can be adjusted depending on the specific problem scale and requirements. For instance, increasing the number of generations may improve solution quality for large instances, while a higher mutation rate can enhance population diversity and help escape local optima.

Table 7.1 summarizes the key simulation parameters used across all experiments. GNs are generated using two PCP parameter sets: set (a) for fronthaul evaluation produces more GNs than set (b) for backhaul evaluation.

Table 7.1 Simulation Parameters

Parameter	**Value**
DBS height (h_0), MBS height, environment variables (α, β)	60 m, 25 m, 9.61, 0.16 [38]
Expected path loss (η_{LOS}, η_{NLOS}), carrier frequency (f_c), GN traffic load (R_n), number of MBSs (B)	1 dB, 20 dB [38], 2 GHz, 20 Mbps, 4
Beam waist radius (ω_0), weather attenuation coefficient (κ), FSO transmit power (P_{FSO})	0.25 cm, 4.3×10^{-4} m^{-1}, 50 mW [43]
Wavelength (λ_0), lens radius (r_0), photodetector responsivity (η_{FSO}), noise power (σ_{F}^2)	1550 nm, 0.1 m, 0.5, -60.1 dBm [43]
PCP cluster standard deviation, cluster density, number of GNs per cluster	100, 10^{-6}, 20
GA parameters: max generations, initial population size, crossover rate (χ_{CO}), mutation rate (χ_{M}), elitism rate (χ_{E}), parent selection, saturation stop threshold	400, 400, 0.3, 0.2, 0.1, tournament, 20

7.6.1 Comparison of Different GA Settings

This section evaluates the performance of the proposed GA under various configurations (i.e., combinations of fitness functions and penalty mechanisms) across a limited number of iterations. The results are compared with a baseline random sampling approach that exhaustively explores $N_{\text{exact}} = 10^7$ possible random solutions in the search space. To focus solely on the backhaul optimization aspect, the access layer is neglected by setting $R_{\text{A}}^0 = \infty$, and the number of MBSs is fixed to $N_{\text{B}} = 2$.

Probability of Finding a Correct Solution

Figure 7.3 presents the probability of reaching a valid solution before the stopping condition, plotted for different values of the DBS count M. For small values of M, all GA variants perform similarly to the random sampling baseline. However, as M increases, the performance gap widens. Beyond 30 DBSs, differences between GA configurations become evident, stabilizing around 40 with a 13% success rate gap between the best (NVP) and worst (NNP) strategies.

Similarly, Figure 7.4 shows the probability of success across different backhaul distance constraints $d_{\max}$. The most challenging cases occur for short distances (e.g., 2.0 km). Above 2.4 km, differences across GA variants diminish, and beyond 2.8 km, they converge. Notably, the NVP strategy achieves the highest success rate for stringent distance constraints, whereas the NNP and random sampling perform poorly.

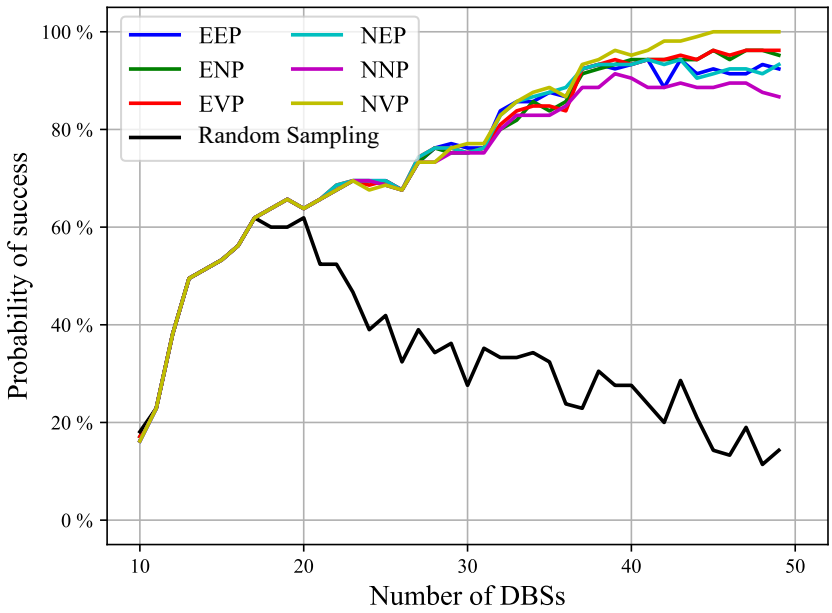


Figure 7.3 Probability of success vs. number of DBSs, M, for different GA configurations and the baseline method. The NVP strategy yields the highest success rates for large M.

Quality of Achieved Correct Solutions

The quality of valid solutions is evaluated using the fitness metric $F_{\text{node}}(\mathcal{S})$ defined in (7.13), which captures the cumulative backhaul load surplus across nodes. This surplus represents

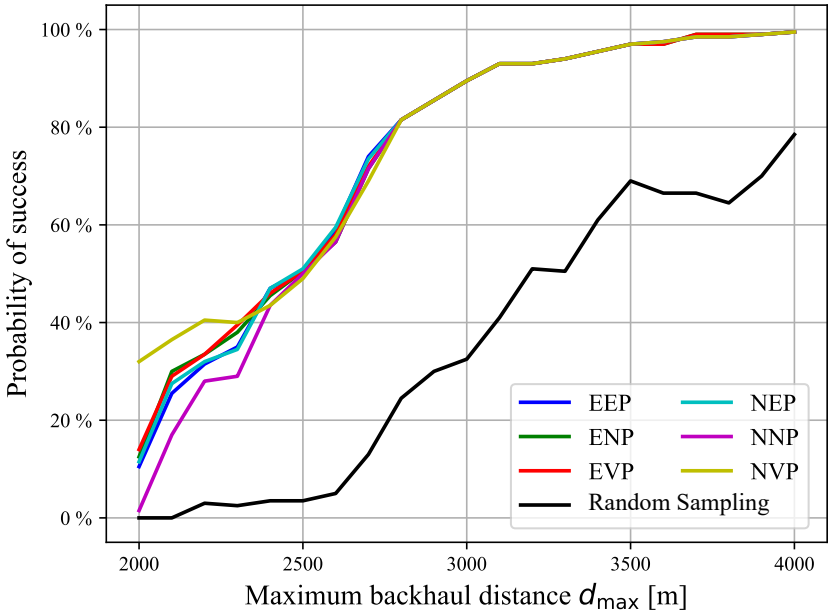


Figure 7.4 Probability of success vs. backhaul distance $d_{\max}$ for different GA configurations and the baseline method. NVP dominates under short backhaul distances.

the system's resilience to increased traffic demands, which helps in avoiding additional DBS deployment.

Figure 7.5 shows that the quality improves as M increases, with identical results across algorithms for small values of M (up to 15). As the network size grows, GA variants diverge: ENP outperforms others, while NVP yields the lowest throughput reserves. Similarly, in Figure 7.6, the ENP variant produces the highest quality results under constrained $d_{\max}$, while NVP lags behind.

The substantially higher success rate of the NVP fitness function arises from its strong emphasis on locating feasible solutions. This stems from the fact that NVP favors surplus at nodes (fitness functions with codes beginning with N, namely NNP, NEP, and NVP), in contrast to edge-focused approaches (codes beginning with E, such as ENP, EEP, and EVP), even though a solution's validity ultimately depends on ensuring sufficient backhaul capacity at each node. Moreover, the Value Penalty mechanism (codes ending in VP—i.e., EVP and NVP) applies a strict penalty that forces all incorrect solutions to yield negative values. This contrasts with the Edge Penalty strategy (codes ending in EP—i.e., EEP and NEP), where penalties may not consistently produce negative results, and the No Penalty setting (codes ending in NP—i.e., ENP and NNP), which applies no penalty at all. These combined

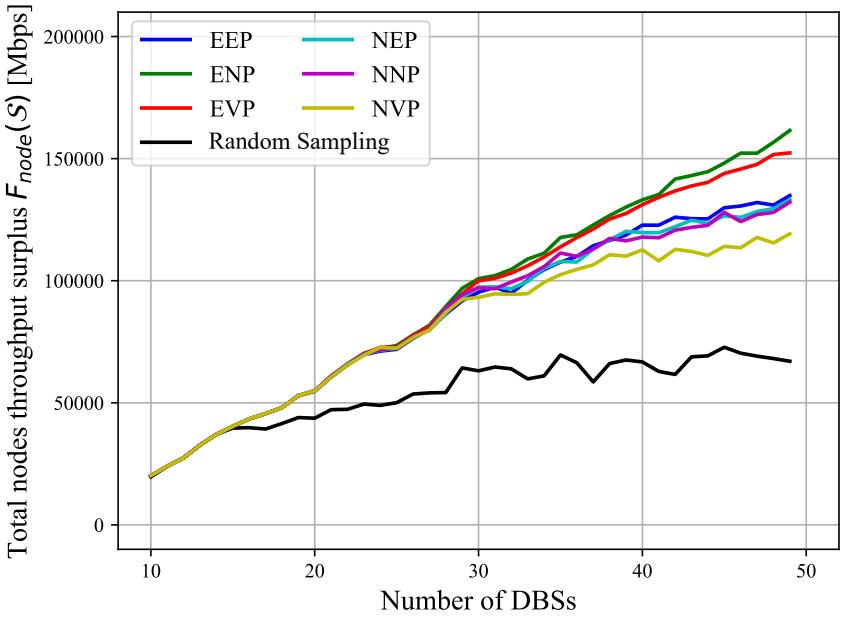


Figure 7.5 Total throughput surplus $F_{\text{node}}(\mathcal{S})$ vs. number of DBSs, M, for different GA configurations and the baseline. ENP yields the highest quality for large M.

characteristics contribute to the high likelihood of discovering a feasible solution, although potentially at the expense of bypassing paths that might yield more optimal results.

Conversely, the ENP (Edge No Penalty) fitness function evaluates solutions purely based on surplus at the edge level and applies no penalty for invalid configurations. As reflected in the results, this approach allows the exploration of less constrained regions of the solution space, which may lead to lower overall success rates but enables the discovery of high-quality, valid solutions, evidenced by greater node throughput surplus.

In summary, NVP is more suitable when the existence of a feasible solution is uncertain, as it maximizes the probability of finding one. In contrast, ENP proves advantageous in well-provisioned scenarios where, for example, there is high drone density, and where feasibility is more likely, allowing the algorithm to emphasize performance optimization.

7.6.2 GA Success Probability: HC vs. K-means Placement

Figure 7.7 compares the performance of the proposed NVP version of the GA, following placement determined either by K-means or by the HC algorithm under different N_{B} settings. Since there is no way for K-means to conserve the coverage radius constraint, the author set $R_{\text{A}}^{0} = \infty$. The performance of both strategies is similar. With a relaxed $d_{\max}$ of 3.5 km,

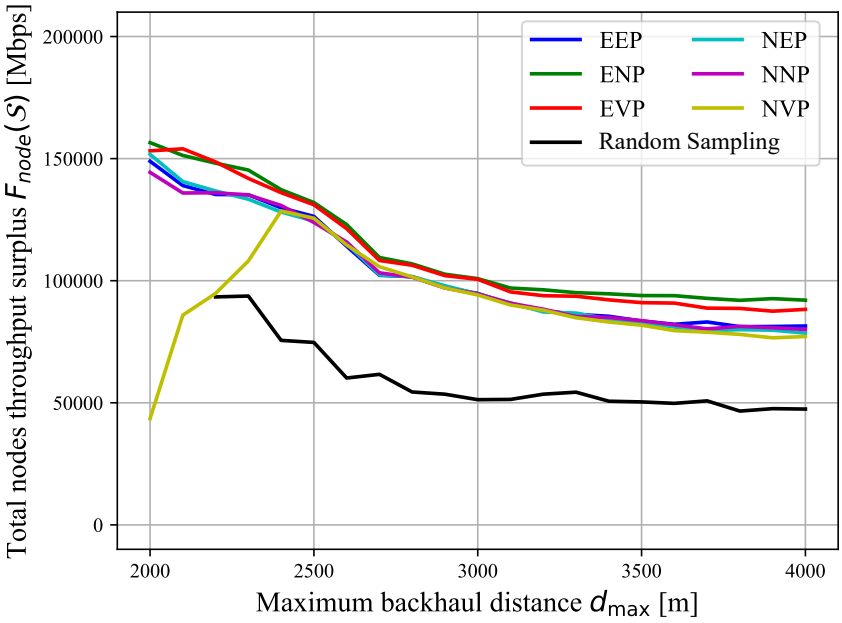


Figure 7.6 Total throughput surplus $F_{\text{node}}(\mathcal{S})$ vs. $d_{\max}$ for different GA configurations and the baseline. ENP shows best performance for constrained distances.

K-means seems to be performing slightly better at the smallest value of M, where it seems that in some cases fewer MBSs are connected to the resulting graph when HC is applied. On the other hand, for a more stringent $d_{\max}$ of 2.5 km, two regions can be identified. When $20 \leq M \leq 35$, increasing the minimum number of required neighbors improves the GA success probability, and HC outperforms K-means in all cases. As the solution space grows larger with more DBSs, however, this improvement diminishes, and the GA's success probability decreases for the specified limited number of generations and initial population size. In such cases, the number of GA generations or the starting population size should be increased.

7.6.3 GA Execution Time

Figure 7.8 shows the average execution time per run. For the baseline random sampling approach, results are reported only for $M + B < 15$ due to a time cap of 10 seconds per instance. The trend line for GA exhibits a quadratic growth, consistent with the theoretical complexity of $O((M+B)^2)$ derived in (7.14).

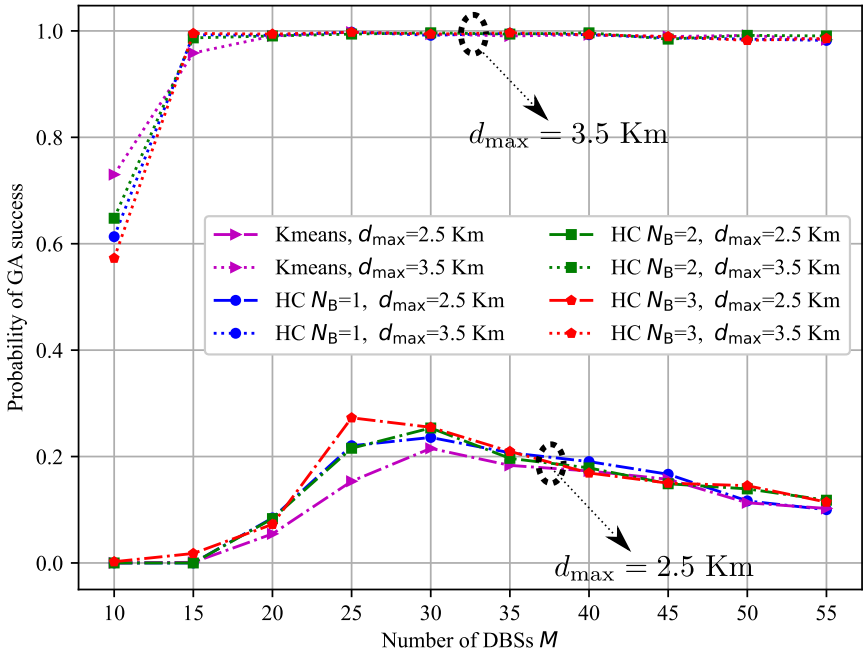


Figure 7.7 GA success probability vs. number of DBSs, M, under different $d_{\max}$ values. Comparison is made between HC and K-means placement strategies.

7.7 Conclusion

This chapter presented a comprehensive framework for designing a joint fronthaul and backhaul network architecture for DBS-assisted wireless networks. In this framework, and leveraging their altitude, DBSs rely on multi-hop wireless links for backhaul connectivity such as FSO. To address the resulting challenges in topology design, the DNP is formulated as a constrained graph optimization problem and demonstrates its NP-hardness.

To solve the problem efficiently, a tailored GA is introduced that results in feasible backhaul paths while maximizing throughput redundancy. This included a novel genome representation and designed two custom fitness functions along with penalty strategies to guide solution evolution. The proposed algorithm ensures flow conservation at every DBS and adheres to connectivity constraints.

Simulation results demonstrated the effectiveness of the algorithm in identifying high-quality solutions under diverse settings. The analysis showed that the NVP fitness configuration yields the highest success probability, especially under strict constraints, while the ENP variant consistently achieves the best throughput performance. The proposed GA scales well with problem size, offering a computationally tractable method for complex backhaul mesh design.

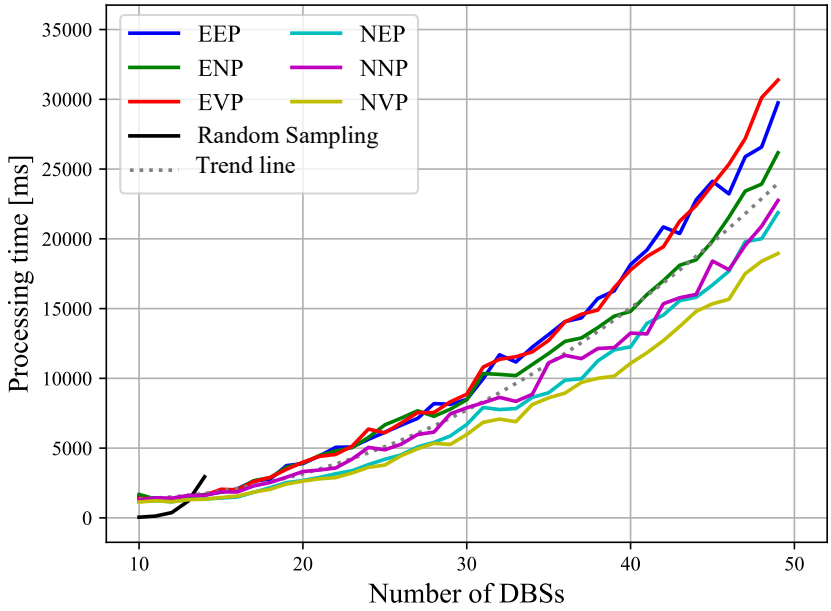


Figure 7.8 Processing time vs. DBSs count M for different GA settings and the baseline method. The quadratic complexity $O((M+B)^2)$ is confirmed.

Furthermore, the author integrated the backhaul optimization into the full network framework introduced in the previous chapter by coupling it with the HC-based DBS placement strategy. The complete framework adapts to changes in user distribution or traffic load, ensuring resilient and efficient end-to-end network connectivity.

Chapter 8

Conclusions

This thesis explored the design of next-generation wireless networks supported by DBSs, which offer flexible, adaptive, and cost-effective connectivity solutions in diverse deployment scenarios. By leveraging the unique spatial and mobility characteristics of UAVs, several critical challenges related to their deployment and operation were addressed, focusing on fronthaul and backhaul network design, energy constraints, and adaptive placement strategies.

The work began by reviewing the classification of UAVs, regulatory constraints, and technical challenges associated with DBS deployment. A clear research scope was established, focusing on aspects unique to airborne networks, namely 3D placement, power control, energy harvesting, and wireless backhaul provisioning.

Then the thesis introduced different simulation frameworks that enabled the modeling of GN distribution and mobility, urban environments, wireless channels, and energy consumption and harvesting aspects. These frameworks provide a guideline for studying such networks and for future work targeting similar problems.

The core contributions of this thesis are organized into five algorithmic modules across Chapters 3–7. These contributions can be summarized as follows:

- **Fronthaul-only DBS placement:** Two placement schemes were proposed—one based on a novel SINR-guided EM clustering method for known GN locations, and another leveraging a multi-agent Q-learning framework for placement in unknown user distributions. Both solutions demonstrated improved coverage, energy efficiency, and fairness. A height optimization technique is introduced, which improves the SNR and can be used to extend the coverage radius when GNs leave the cell boundary. Furthermore, an access probability metrics was derived to evaluate the performance of a beamforming DBS which is covering vehicles under a single beam. This metric was

also used within an MILP formulation to determine the placement of DBSs and their beams footprints, while minimizing the number of active DBSs.

- **Transmission power optimization:** An MCTS-based (TP-MCTS) framework was developed to optimize DBS transmission power settings. The method improved energy efficiency and link reliability and could deactivate redundant drones by assigning zero power.
- **Backhaul-aware placement:** A visibility-graph framework was proposed for LOS-constrained DRSs deployment in urban environments to relay backhaul traffic. This model is extended to incorporate solar energy harvesting and RIS-based vitrual links.
- **Joint fronthaul-backhaul placement:** An HC algorithm was formulated to jointly satisfy GN coverage and mesh connectivity constraints, providing reliable coverage while maintaining backhaul robustness through an N_{B}-neighbor constraint. It was also shown how the placement can adapt to GNs through height optimization.
- **Backhaul mesh topology design:** The thesis introduced a tailored GA to solve the DNP under mixed-integer linear constraints. The proposed algorithm ensures flow conservation and maximizes redundant throughput across the mesh, and integrates with the HC-based placement scheme to provide a comprehensive deplyment framework for DBS-assisted networks.

The author demonstrated that intelligent AI-based optimization can transform DBSs into resilient and autonomous 3D networks, thus confirming the thesis claim.

8.1 Future Work

Several extensions are possible based on this thesis. First, integrating time-domain scheduling and frequency planning with spatial placement could further enhance performance. Second, incorporating user mobility prediction into the placement algorithms could improve service continuity. Third, the backhaul optimization frameworks can be extended to include further aspects such as latency. Fourth, the comprehensive framework introduced in the final chapter can be furtuer optimized by leveraging GA and HC historical records when adapting to the dynamic changes of GNs throughput and locations, thus reducing the complexity by avoiding the need for complete network recomputation. Finally, real-world field tests and hardware-in-the-loop simulations will be essential for validating the practical applicability of the proposed solutions.

Bibliography

[1] Y. Zeng, Q. Wu, and R. Zhang, "Accessing From the Sky: A Tutorial on UAV Communications for 5G and Beyond," *Proceedings of the IEEE*, vol. 107, no. 12, pp. 2327–2375, Dec. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8918497/

[2] A. Fotouhi, H. Qiang, M. Ding, M. Hassan, L. G. Giordano, A. Garcia-Rodriguez, and J. Yuan, "Survey on UAV Cellular Communications: Practical Aspects, Standardization Advancements, Regulation, and Security Challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3417–3442, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8675384/

[3] A. F. M. Shahen Shah, "A survey from 1g to 5g including the advent of 6g: Architectures, multiple access techniques, and emerging technologies," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, pp. 1117–1123.

[4] J. Qiu, D. Grace, G. Ding, M. D. Zakaria, and Q. Wu, "Air-ground heterogeneous networks for 5g and beyond via integrating high and low altitude platforms," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 140–148, 2019.

[5] M. Dai, N. Huang, Y. Wu, J. Gao, and Z. Su, "Unmanned-Aerial-Vehicle-Assisted Wireless Networks: Advancements, Challenges, and Solutions," *IEEE Internet of Things Journal*, vol. 10, no. 5, pp. 4117–4147, Mar. 2023. [Online]. Available: https://ieeexplore.ieee.org/document/9994654/

[6] E. U. A. S. A. (EASA). (2024, Apr.) Open Category - Low Risk - Civil Drones. https://www.easa.europa.eu/en/domains/civil-drones-rpas/open-category-civil-drones.

[7] H. Bayerlein, "Machine Learning Methods for UAV-aided Wireless Networks," Theses, Sorbonne Université, Sep. 2021. [Online]. Available: https://theses.hal.science/tel-03465118

[8] E. U. A. S. A. (EASA). (2024, Apr.) Operating a drone. https://www.easa.europa.eu/en/domains/drones-air-mobility/operating-drone.

[9] L. Zhang, H. Zhao, S. Hou, Z. Zhao, H. Xu, X. Wu, Q. Wu, and R. Zhang, "A Survey on 5G Millimeter Wave Communications for UAV-Assisted Wireless Networks," *IEEE Access*, vol. 7, pp. 117 460–117 504, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8764406/

[10] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8660516/

[11] B. Li, Z. Fei, and Y. Zhang, "UAV Communications for 5G and Beyond: Recent Advances and Future Trends," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, Apr. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8579209/

[12] C. T. Cicek, H. Gultekin, B. Tavli, and H. Yanikomeroglu, "UAV Base Station Location Optimization for Next Generation Wireless Networks: Overview and Future Research Directions," in *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)*. Muscat, Oman: IEEE, Feb. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8658363/

[13] B. Tezergil and E. Onur, "Wireless backhaul in 5G and beyond: Issues, challenges and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2579–2632, 2022.

[14] M. Lu, M. Bagheri, A. P. James, and T. Phung, "Wireless Charging Techniques for UAVs: A Review, Reconceptualization, and Extension," *IEEE Access*, vol. 6, pp. 29 865–29 884, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8368105/

[15] B. Michini, T. Toksoz, J. Redding, M. Michini, J. How, M. Vavrina, and J. Vian, "Automated Battery Swap and Recharge to Enable Persistent UAV Missions," in *Infotech@Aerospace 2011*. St. Louis, Missouri: American Institute of Aeronautics and Astronautics, Mar. 2011. [Online]. Available: https://arc.aiaa.org/doi/10.2514/6.2011-1405

[16] M. C. Achtelik, J. Stumpf, D. Gurdan, and K.-M. Doth, "Design of a flexible high performance quadcopter platform breaking the MAV endurance record with laser power beaming," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA: IEEE, Sep. 2011, pp. 5166–5172. [Online]. Available: http://ieeexplore.ieee.org/document/6094731/

[17] J. Moore and R. Tedrake, "Magnetic localization for perching UAVs on powerlines," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. San Francisco, CA: IEEE, Sep. 2011, pp. 2700–2707. [Online]. Available: http://ieeexplore.ieee.org/document/6095188/

[18] E. J. Silberg and J. H. Milgram, "Battery charging arrangement for unmanned aerial vehicle utilizing the electromagnetic field associated with utility power lines to generate power to inductively charge energy supplies," May 11 2010, uS Patent 7,714,536.

[19] A. B. Junaid, Y. Lee, and Y. Kim, "Design and implementation of autonomous wireless charging station for rotary-wing UAVs," *Aerospace Science and Technology*, vol. 54, pp. 253–266, Jul. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1270963816301547

[20] M. Simic, C. Bil, and V. Vojisavljevic, "Investigation in Wireless Power Transmission for UAV Charging," *Procedia Computer Science*, vol. 60, pp. 1846–1855, 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1877050915024229

[21] S. Jashnani, T. Nada, M. Ishfaq, A. Khamker, and P. Shaholia, "Sizing and preliminary hardware testing of solar powered UAV," *The Egyptian Journal of Remote Sensing and Space Science*, vol. 16, no. 2, pp. 189–198, Dec. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1110982313000112

[22] S. Morton, R. D'Sa, and N. Papanikolopoulos, "Solar powered UAV: Design and experiments," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Hamburg, Germany: IEEE, Sep. 2015, pp. 2460–2466. [Online]. Available: http://ieeexplore.ieee.org/document/7353711/

[23] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, K. Alexis, and R. Siegwart, "Perpetual flight with a small solar-powered UAV: Flight results, performance analysis and model validation," in *2016 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2016, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/document/7500855/

[24] S. Sekander, H. Tabassum, and E. Hossain, "Statistical Performance Modeling of Solar and Wind-Powered UAV Communications," *IEEE Transactions on Mobile Computing*, vol. 20, no. 8, pp. 2686–2700, Aug. 2021, https://ieeexplore.ieee.org/document/9060991/.

[25] D. Liu, L. Wang, Y. Chen, M. Elkashlan, K.-K. Wong, R. Schober, and L. Hanzo, "User Association in 5G Networks: A Survey and an Outlook," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1018–1044, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7378276/

[26] Y. Xu, G. Gui, H. Gacanin, and F. Adachi, "A Survey on Resource Allocation for 5G Heterogeneous Networks: Current Research, Future Trends, and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 668–695, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9355403/

[27] M. Mozaffari, A. Taleb Zadeh Kasgari, W. Saad, M. Bennis, and M. Debbah, "Beyond 5G With UAVs: Foundations of a 3D Wireless Cellular Network," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 357–372, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8533634/

[28] C. Saha, M. Afshang, and H. S. Dhillon, "Poisson cluster process: Bridging the gap between PPP and 3GPP HetNet models," in *2017 Information Theory and Applications Workshop (ITA)*, 2017, pp. 1–9.

[29] M. Mirahsan, R. Schoenen, and H. Yanikomeroglu, "Hethetnets: Heterogeneous traffic distribution in heterogeneous wireless cellular networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2252–2265, 2015.

[30] C. Saha, M. Afshang, and H. S. Dhillon, "Enriched k -tier hetnet model to enable the analysis of user-centric small cell deployments," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1593–1608, 2017.

[31] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang, "A group mobility model for ad hoc wireless networks," in *Proceedings of the 2nd ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 53–60. [Online]. Available: https://doi.org/10.1145/313237.313248

[32] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Eds. Boston, MA: Springer US, 1996, pp. 153–181. [Online]. Available: https://doi.org/10.1007/978-0-585-29603-6_5

[33] A. Jardosh, E. M. Belding-Royer, K. C. Almeroth, and S. Suri, "Towards realistic mobility models for mobile ad hoc networks," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '03. New York, NY, USA: Association for Computing Machinery, 2003, p. 217–229. [Online]. Available: https://doi.org/10.1145/938985.939008

[34] "OpenStreetMap." [Online]. Available: https://www.openstreetmap.org/

[35] P. Agyapong, V. Braun, M. Fallgren, A. Gouraud, M. Hessler, S. Jeux, A. Klein, J. Lianghai, D. Martín-Sacristán, M. Maternia, M. Moisio, J. F. Monserrat, K. Pawlak, H. Tullberg, and A. Weber, "ICT-317669-METIS/D6.1 Simulation Guidelines," Oct. 2013. [Online]. Available: https://cordis.europa.eu/docs/projects/cnect/9/317669/080/deliverables/001-METISD61v1pdf.pdf

[36] C. Yan, L. Fu, J. Zhang, and J. Wang, "A Comprehensive Survey on UAV Communication Channel Modeling," *IEEE Access*, vol. 7, pp. 107 769–107 792, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8787874/

[37] A. A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, and P. Dobbins, "A Survey of Channel Modeling for UAV Communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 2804–2821, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8411465/

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

[39] ITU, "Recommendation ITU-R P.1410-6 (08/2023) - Propagation data and prediction methods required for the design of terrestrial broadband radio access systems operating in a frequency range from 3 GHz to 60 GHz," *P Series*, vol. Radiowave Propagation, Geneva, Switzerland, Rec. P.1410-2, 2003. [Online]. Available: https://www.itu.int/rec/R-REC-P.1410-6-202308-I/en

[40] Hyuck Kwon and T. Birdsall, "Channel capacity in bits per joule," *IEEE Journal of Oceanic Engineering*, vol. 11, no. 1, pp. 97–99, Jan. 1986. [Online]. Available: http://ieeexplore.ieee.org/document/1145138/

[41] E. Lee, J. Park, D. Han, and G. Yoon, "Performance analysis of the asymmetric dual-hop relay transmission with mixed RF/FSO links," *IEEE Photonics Technology Letters*, vol. 23, no. 21, pp. 1642–1644, 2011.

[42] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir, "Broadband Millimeter-Wave Propagation Measurements and Models Using Adaptive-Beam Antennas for Outdoor Urban Cellular Communications," *IEEE Transactions on Antennas and Propagation*, vol. 61, no. 4, pp. 1850–1859, Apr. 2013. [Online]. Available: https://ieeexplore.ieee.org/document/6387266/

[43] M. e. a. Najafi, "Statistical modeling of the FSO fronthaul channel for UAV-based communications," *IEEE Transactions on Communications*, vol. 68, no. 6, pp. 3720–3736, 2020.

[44] Y. e. a. Kaymak, "A survey on acquisition, tracking, and pointing mechanisms for mobile free-space optical communications," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1104–1123, 2018.

[45] M. A. Kishk and M.-S. Alouini, "Exploiting Randomly Located Blockages for Large-Scale Deployment of Intelligent Surfaces," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 4, pp. 1043–1056, Apr. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9174910/

[46] X. Liu, Y. Liu, and Y. Chen, "Machine Learning Empowered Trajectory and Passive Beamforming Design in UAV-RIS Wireless Networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2042–2055, Jul. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9277627/

[47] Y. Sun, D. Xu, D. W. K. Ng, L. Dai, and R. Schober, "Optimal 3D-trajectory design and resource allocation for solar-powered UAV communication systems," *IEEE Transactions on Communications*, vol. 67, no. 6, pp. 4281–4298, 2019.

[48] S. Dimitriou, S. Kapsalis, P. Panagiotou, and K. Yakinthos, "Energy efficiency improvement potential of a tactical BWB UAV using renewable energy sources," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2020, pp. 1384–1391.

[49] R. Singh, T. Jaakkola, and A. Mohammad, "6.867 machine learning. fall 2006." [Online]. Available: MITOpenCourseWare,https://ocw.mit.edu.

[50] D. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, 2021.

[51] M.-A. Lahmeri, M. A. Kishk, and M.-S. Alouini, "Artificial Intelligence for UAV-Enabled Wireless Networks: A Survey," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 1015–1040, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9411810/

[52] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[53] M. Alzenad, A. El-Keyi, and H. Yanikomeroglu, "3-D placement of an unmanned aerial vehicle base station for maximum coverage of users with different QoS requirements," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 38–41, 2018.

[54] R. I. Bor-Yaliniz, A. El-Keyi, and H. Yanikomeroglu, "Efficient 3-D placement of an aerial base station in next generation cellular networks," in *2016 IEEE International Conference on Communications (ICC)*. Kuala Lumpur, Malaysia: IEEE, May 2016, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/7510820/

[55] J. Lyu, Y. Zeng, R. Zhang, and T. J. Lim, "Placement optimization of UAV-mounted mobile base stations," *IEEE Communications Letters*, vol. 21, no. 3, pp. 604–607, 2017.

[56] H. Huang and A. V. Savkin, "A method for optimized deployment of unmanned aerial vehicles for maximum coverage and minimum interference in cellular networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2638–2647, 2019.

[57] H. V. Abeywickrama, Y. He, E. Dutkiewicz, B. A. Jayawickrama, and M. Mueck, "A reinforcement learning approach for fair user coverage using UAV mounted base stations under energy constraints," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 67–81, 2020.

[58] D. M. Chiu, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," Digital Equipment Corporation, Tech. Rep., 1984.

[59] J. Qiu, J. Lyu, and L. Fu, "Placement optimization of aerial base stations with deep reinforcement learning," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, 2020, pp. 1–6.

[60] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, 2018.

[61] M. S. Daskin and K. L. Maass, "The p-Median Problem," in *Location Science*, G. Laporte, S. Nickel, and F. Saldanha Da Gama, Eds. Cham: Springer International Publishing, 2015, pp. 21–45. [Online]. Available: http://link.springer.com/10.1007/978-3-319-13111-5_2

[62] N. Megiddo and K. J. Supowit, "On the Complexity of Some Common Geometric Location Problems," *SIAM Journal on Computing*, vol. 13, no. 1, pp. 182–196, Feb. 1984. [Online]. Available: http://epubs.siam.org/doi/10.1137/0213014

[63] V. E. Brimkov, A. Leach, J. Wu, and M. Mastroianni, "Approximation algorithms for a geometric set cover problem," *Discrete Applied Mathematics*, vol. 160, no. 7, pp. 1039–1052, 2012.

[64] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[65] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Algorithm for energy efficient inter-UAV collision avoidance," in *2017 17th International Symposium on Communications and Information Technologies (ISCIT)*, 2017, pp. 1–5.

[66] "14 CFR Part 107 – Small Unmanned Aircraft Systems." [Online]. Available: https://www.ecfr.gov/current/title-14/part-107

[67] "Géoportail." [Online]. Available: https://www.geoportail.gouv.fr/

[68] A. Al-Hourani, "Interference modeling in low-altitude unmanned aerial vehicles," *IEEE Wireless Communications Letters*, vol. 9, no. 11, pp. 1952–1955, 2020.

[69] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2007, p. 1027–1035.

[70] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018. [Online]. Available: http://incompleteideas.net/book/the-book-2nd.html

[71] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *J. Mach. Learn. Res.*, vol. 10, p. 1633–1685, dec 2009.

[72] M. H. C. Garcia, A. Molina-Galan, M. Boban, J. Gozalvez, B. Coll-Perales, T. Şahin, and A. Kousaridas, "A tutorial on 5g nr v2x communications," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1972–2026, 2021.

[73] F. Conserva and R. Verdone, "Analytical description of access probability and rra strategy for uav-aided vehicular applications," in *Proceedings of the Ninth Workshop on Micro Aerial Vehicle Networks, Systems, and Applications*, ser. DroNet '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 21–26. [Online]. Available: https://doi.org/10.1145/3597060.3597241

[74] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of Monte Carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, 2012.

[75] S. Chen, X. Li, C. Luo, H. Ji, and H. Zhang, "Energy-efficient power, position and time control in uav-assisted wireless networks," in *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.

[76] S. Zhang, S. Shi, S. Gu, and X. Gu, "Power control and trajectory planning based interference management for uav-assisted wireless sensor networks," *IEEE Access*, vol. 8, pp. 3453–3464, 2020.

[77] G. Zhang, Q. Wu, M. Cui, and R. Zhang, "Securing uav communications via joint trajectory and power control," *IEEE Transactions on Wireless Communications*, vol. 18, no. 2, pp. 1376–1389, 2019.

[78] L. Li, T.-H. Chang, and S. Cai, "Uav positioning and power control for two-way wireless relaying," *IEEE Transactions on Wireless Communications*, vol. 19, no. 2, pp. 1008–1024, 2020.

[79] P. Auer, "Using upper confidence bounds for online learning," in *Proceedings 41st Annual Symposium on Foundations of Computer Science*, 2000, pp. 270–279.

[80] E. Kalantari, H. Yanikomeroglu, and A. Yongacoglu, "On the number and 3D placement of drone base stations in wireless cellular networks," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–6.

[81] Y. He, W. Ma, and J. Zhang, "The parameters selection of PSO algorithm influencing on performance of fault diagnosis," *MATEC Web of Conferences*, vol. 63, p. 02019, 01 2016.

[82] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 2, no. 3, pp. 257–269, 2003.

[83] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?" *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[84] U. Siddique, H. Tabassum, E. Hossain, and D. I. Kim, "Wireless backhauling of 5G small cells: Challenges and solution approaches," *IEEE Wireless Communications*, vol. 22, no. 5, pp. 22–31, Oct. 2015, http://ieeexplore.ieee.org/document/7306534/.

[85] A. U. Chaudhry and H. Yanikomeroglu, "Free Space Optics for Next-Generation Satellite Networks," *IEEE Consumer Electronics Magazine*, vol. 10, no. 6, pp. 21–31, Nov. 2021, https://ieeexplore.ieee.org/document/9219130/.

[86] A. Kaadan, H. H. Refai, and P. G. LoPresti, "Multielement FSO Transceivers Alignment for Inter-UAV Communications," *Journal of Lightwave Technology*, vol. 32, no. 24, pp. 4785–4795, Dec. 2014, http://ieeexplore.ieee.org/document/6936323/.

[87] M. Alzenad, M. Z. Shakir, H. Yanikomeroglu, and M.-S. Alouini, "FSO-Based Vertical Backhaul/Fronthaul Framework for 5G+ Wireless Networks," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 218–224, Jan. 2018, http://ieeexplore.ieee.org/document/8255764/.

[88] H. Ajam, M. Najafi, V. Jamali, and R. Schober, "Ergodic Sum Rate Analysis of UAV-Based Relay Networks With Mixed RF-FSO Channels," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 164–178, 2020, https://ieeexplore.ieee.org/document/8970423/.

[89] W. Fawaz, C. Abou-Rjeily, and C. Assi, "UAV-Aided Cooperation for FSO Communication Systems," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 70–75, Jan. 2018, https://ieeexplore.ieee.org/document/8255740/.

[90] Y. Zeng and R. Zhang, "Energy-Efficient UAV Communication With Trajectory Optimization," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017, http://ieeexplore.ieee.org/document/7888557/.

[91] D. Lee, *Proximity and Reachability in the Plane*. University of Illinois at Urbana-Champaign, 1978, https://books.google.pl/books?id=v3vAtgAACAAJ.

[92] J. A. Del Peral-Rosado, J. M. Parro-Jimenez, J. A. Lopez-Salcedo, G. Seco-Granados, P. Crosta, F. Zanier, and M. Crisci, "Comparative results analysis on positioning with real LTE signals and low-cost hardware platforms," in *2014 7th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*. Noordwijk, Netherlands: IEEE, Dec. 2014, pp. 1–8, http://ieeexplore.ieee.org/document/7045148/.

[93] Y. Liu, X. Wang, G. Boudreau, A. B. Sediq, and H. Abou-zeid, "Deep Learning Based Hotspot Prediction and Beam Management for Adaptive Virtual Small Cell in 5G Networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 1, pp. 83–94, Feb. 2020, https://ieeexplore.ieee.org/document/8968715/.

[94] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communications of the ACM*, vol. 22, no. 10, pp. 560–570, 1979.

[95] D. Coleman, "Lee's O (n2 log n) visibility graph algorithm implementation and analysis," *Department of Computer Science, University of Colorado at Boulder: Boulder, CO, USA*, 2012.

[96] H.-B. Jeon, S.-H. Park, J. Park, K. Huang, and C.-B. Chae, "RIS-assisted Aerial Backhaul System for UAV-BSs: An Energy-efficiency Perspective," in *2021 IEEE Global Communications Conference (GLOBECOM)*. Madrid, Spain: IEEE, Dec. 2021, pp. 1–6, https://ieeexplore.ieee.org/document/9685565/.

[97] E. M. Mohamed, M. Alnakhli, S. Hashima, and M. Abdel-Nasser, "Distribution of Multi MmWave UAV Mounted RIS Using Budget Constraint Multi-Player MAB," *Electronics*, vol. 12, no. 1, p. 12, Dec. 2022, https://www.mdpi.com/2079-9292/12/1/12.

[98] P. Mogensen, W. Na, I. Z. Kovacs, F. Frederiksen, A. Pokhariyal, K. I. Pedersen, T. Kolding, K. Hugl, and M. Kuusela, "LTE Capacity Compared to the Shannon Bound," in *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*. Dublin, Ireland: IEEE, Apr. 2007, pp. 1234–1238, http://ieeexplore.ieee.org/document/4212688/.

[99] B. M. et al., "Demonstration of free-space optical communication for long-range data links between balloons on Project Loon," in *Free-Space Laser Communication and Atmospheric Propagation XXIX*, vol. 10096, International Society for Optics and Photonics. SPIE, 2017, p. 100960Z.

[100] "Meta Connectivity." [Online]. Available: https://www.facebook.com/connectivity/solutions/free-basics

[101] S. M. Mohamed, H. S. Hamza, and I. A. Saroit, "Coverage in mobile wireless sensor networks (m-wsn): A survey," *Computer Communications*, vol. 110, pp. 133–150, 2017.

[102] S. M. Taleb, Y. Meraihi, A. B. Gabis, S. Mirjalili, and A. Ramdane-Cherif, "Nodes placement in wireless mesh networks using optimization approaches: a survey," *Neural Computing and Applications*, vol. 34, no. 7, pp. 5283–5319, Apr. 2022.

[103] C. Qiu, Z. Wei, X. Yuan, Z. Feng, and P. Zhang, "Multiple UAV-mounted base station placement and user association with joint fronthaul and backhaul optimization," *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5864–5877, 2020.

[104] A. Alzidaneen, A. Alsharoa, and M.-S. Alouini, "Resource and placement optimization for multiple UAVs using backhaul tethered balloons," *IEEE Wireless Communications Letters*, vol. 9, no. 4, pp. 543–547, 2020.

[105] M. Nikooroo, O. Esrafilian, Z. Becvar, and D. Gesbert, "Sum capacity maximization in multi-hop mobile networks with flying base stations," in *IEEE GLOBECOM*, 2022, pp. 6565–6570.

[106] M. K. Shehzad, A. Ahmad, S. A. Hassan, and H. Jung, "Backhaul-aware intelligent positioning of UAVs and association of terrestrial base stations for fronthaul connectivity," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 4, pp. 2742–2755, 2021.

[107] F. Xhafa, C. Sanchez, and L. Barolli, "Ad hoc and neighborhood search methods for placement of mesh routers in wireless mesh networks," in *29th IEEE International Conference on Distributed Computing Systems Workshops*, 2009, pp. 400–405.

[108] A. A. Franklin and C. S. R. Murthy, "Node placement algorithm for deployment of two-tier wireless mesh networks," in *IEEE GLOBECOM - IEEE Global Telecommunications Conference*, 2007, pp. 4823–4827.

[109] E. Amaldi, A. Capone, M. Cesana, I. Filippini, and F. Malucelli, "Optimization models and methods for planning wireless mesh networks," *Computer Networks*, vol. 52, no. 11, pp. 2159–2171, Aug. 2008.

[110] C.-C. Chen, C. Chekuri, and D. Klabjan, "Topology formation for wireless mesh network planning," in *IEEE INFOCOM*, 2009, pp. 2671–2675.

[111] S. Poduri and G. Sukhatme, "Constrained coverage for mobile sensor networks," in *IEEE International Conference on Robotics and Automation. Proceedings. ICRA '04.*, vol. 1, 2004, pp. 165–171.

[112] M. Erdelj, T. Razafindralambo, and D. Simplot-Ryl, "Covering points of interest with mobile sensors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 32–43, 2013.

[113] S. Theodoridis and K. Koutroumbas, "Chapter 13 - clustering algorithms II: Hierarchical algorithms," in *Pattern Recognition (Fourth Edition)*, 4th ed. Boston: Academic Press, 2009, pp. 653–700.

[114] M. Gapeyenko, V. Petrov, D. Moltchanov, S. Andreev, N. Himayat, and Y. Koucheryavy, "Flexible and reliable UAV-assisted backhaul operation in 5G mmwave cellular networks," *IEEE JSAC*, vol. 36, no. 11, pp. 2486–2496, 2018.

[115] O. Tipmongkolsilp, S. Zaghloul, and A. Jukan, "The evolution of cellular backhaul technologies: Current issues and future trends," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 1, pp. 97–113, 2011.

[116] U. Challita and W. Saad, "Network formation in the sky: Unmanned aerial vehicles for multi-hop wireless backhauling," in *IEEE GLOBECOM*, 2017, pp. 1–6.

[117] P. Li and J. Xu, "Placement Optimization for UAV-Enabled Wireless Networks with Multi-Hop Backhauls," *Journal of Communications and Information Networks*, vol. 3, no. 4, pp. 64–73, Dec. 2018.

[118] S. Yang, D. Shi, Y. Peng, S. Yang, B. Zhang, and W. Yang, "Placement optimization for UAV-enabled wireless networks with multi-hop backhauls in urban environments," in *21st ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, 2022, pp. 54–66.

[119] E. Odat, H. Ghazzai, and A. Alsharoa, "A WaveGAN approach for mmWave-based FANET topology optimization," *Sensors*, vol. 24, no. 1, 2024.

[120] Q. Liu, X. Jia, and Y. Zhou, "Topology control for multi-channel multi-radio wireless mesh networks using directional antennas," *Wireless Networks*, vol. 17, pp. 41–51, 2011.

[121] J. Sabzehali, V. K. Shah, Q. Fan, B. Choudhury, L. Liu, and J. H. Reed, "Optimizing number, placement, and backhaul connectivity of multi-UAV networks," *IEEE Internet of Things Journal*, vol. 9, no. 21, pp. 21 548–21 560, 2022.

[122] Y. Zhang, M. A. Kishk, and M.-S. Alouini, "Deployment optimization of tethered drone-assisted integrated access and backhaul networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 2668–2680, 2024.

[123] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1979.

[124] Z. Michalewicz and M. Schoenauer, "Evolutionary algorithms for constrained parameter optimization problems," *Evolutionary computation*, vol. 4, no. 1, pp. 1–32, 1996.

[125] A. Gad, "Pygad: an intuitive genetic algorithm python library," *Multimedia Tools and Applications*, vol. 83, pp. 1–14, 12 2023.