Resource Allocation in mmWave 5G IAB Networks: A Reinforcement Learning Approach based on Column Generation

Bibo Zhang[a,*], Francesco Devoti[a,b], Ilario Filippini[a] and Danilo De Donno[c]

[a] Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, 20133 Milan, Italy
[b] NEC Laboratories Europe, Heidelberg, Germany
[c] Milan Research Center, Huawei Technologies Italia S.r.l., 20147 Milan, Italy

ARTICLE INFO

Keywords:
Millimeter-wave Communication
Wireless Access Networks
IAB Networks
Resource Allocation
Deep Reinforcement Learning
Long Short-Term Memory (LSTM)
Column Generation

Abstract

Millimeter wave (mmWave) communications have been introduced in the 5G standardization process due to their attractive potential to provide a huge capacity extension to traditional sub-6 GHz technologies. However, such high-frequency communications are characterized by harsh propagation conditions, thus requiring base stations to be densely deployed. Integrated access and backhaul (IAB) network architecture proposed by 3GPP is gaining momentum as the most promising and cost-effective solution to this need of network densification.

IAB networks' available resources need to be carefully tuned in a complex setting, including directional transmissions, device heterogeneity, and intermittent links with different levels of availability that quickly change over time. It is hard for traditional optimization techniques to provide alone the best performance in these conditions. We believe that Deep Reinforcement Learning (DRL) techniques, especially assisted with Long Short-Term Memory (LSTM), can implicitly capture the regularities of environment dynamics and learn the best resource allocation strategy in networks affected by obstacle blockages. In this article, we propose a DRL based framework based on the Column Generation (CG) that shows remarkable effectiveness in addressing routing and link scheduling in mmWawe 5G IAB networks in realistic scenarios.

1. Introduction

The 3GPP 5G standardization has introduced new frequencies above 24 GHz for Radio Access Networks (RANs), namely Frequency Range 2 (FR2) or millimeter-wave (mmWave) band, as one of the main reliefs from the global mobile traffic growth that is challenging the capacity of access networks with communication technologies below 6 GHz. Large bandwidths (several hundred MHz) available at those mainly-underutilized spectrum portions have unleashed plenty of opportunities to deliver the RAN Gbps-throughput promise.

However, this attractive advantage comes at the cost of a harsh propagation environment characterized by very high path losses and no propagation through obstacles, not only vehicle and buildings but also human bodies. While directional antennas (e.g., phased arrays) can mitigate path losses, although requiring sophisticated hardware and smart beam steering procedures, there is very little they can do against recurrent obstacle blockages. Indeed, mmWave deployments are typically coverage-limited, thus 5G mmWave access networks require closer base stations than traditional radio access networks. This translates into high installation costs for operators that need to connect many sites with fibers.

In order to provide a technically and economically viable solution to the required network densification, 3GPP release 16 specifications have introduced a new multi-hop wireless access architecture, named Integrated Access and Backhaul (IAB)[2]. The idea is to place simpler relay nodes, called IAB-nodes, in the coverage area of a full-fledged mmWave base station (BS), called IAB-donor, and form a wireless backhaul to forward data packets between the IAB-donor and user equipment (UE). The peculiar aspect of this architecture is the self-backhauling approach, where both radio access and wireless backhaul links share the same radio resources and interfaces. Therefore, a proper management of the radio resource allocation is fundamental to operate this network and it is carried out by the IAB-donor. In particular, since the proposed media access control (MAC) solution is based on TDMA, it involves the optimization of the routing paths and the scheduling of directional transmissions along established links.

Routing and scheduling in wireless multi-hop networks have a long-standing literature focusing on optimization techniques that consider always-available links [34, 3, 5]. However, the harsh propagation environment of mmWave frequencies and the strong impact of the obstacles on link availability make these approaches inadequate for mmWave IAB networks. Indeed, the optimal performance provided in ideal link conditions can be hindered by their unpredictable on-off behavior, thus destroying the advantages of the optimization. We could in principle perform an optimization each time the network undergoes a change. However, optimization algorithms are usually time consuming, which makes this solution infeasible for real-time operations.

We believe Reinforcement Learning (RL) techniques are the ideal solution for mmWave IAB networks due to the intrinsic ability of these algorithms to adapt to environment conditions. Indeed, RL agents can be trained to play against the environment to understand what the best strategy is, even

*Corresponding author

bibo.zhang@polimi.it (B. Zhang); francesco.devoti@polimi.it (F. Devoti); ilario.filippini@polimi.it (I. Filippini); danilo.dedonno@huawei.com (D.D. Donno)

ORCID(s):

when the environment's reply is stochastic. We can perform offline training through realistic instances of the actual environment statistics, or we can have an online training and operating system that learns while sending packets to UEs through IAB nodes. Once trained, the agent will be able to play the strategy and provide the best reward in front of any instance of the random environment.

Routing and scheduling in wireless multi-hop networks has been traditionally considered a hard problem due to interference constraints. Solely relying on an RL approach may lead to largely suboptimal working points. This is the reason why we decided to adopt a hybrid approach, which combines traditional optimization techniques and recent RL approaches to synergically provide a quasi-optimal and adaptive resource allocation algorithm.

In this article, we formally introduce the optimization problem of flow routing and link scheduling in mmWave IAB networks, jointly coordinating access and backhaul parts to maximize throughput in a multi-hop network architecture. We solve the problem with a Column Generation (CG)-based approach and leverage its generated variables to populate an optimal candidate action set for the RL agent. Based on these action sets, built of promising scheduling options, we design a Deep Reinforcement Learning (DRL) framework, based on Long Short-Term Memory (LSTM) neural networks, which can overcome the static limits of the optimization approach in dynamic environments. We place emphasis on realistic scenarios and unreliable networks that are vulnerable to recurrent and dynamic blockages. We propose an offline and an online training version of the approach, which we evaluate against traditional approaches via numerical simulations. Furthermore, we discuss feasibility issues in implementing our solution in a real system.

The rest of the paper is organized as follows. We first discuss related works in Section 2 and point out the contributions of this article, then we provide a system overview in Section 3. In Section 4 we present the formulation of the problem based on Column Generation, whose results are used in Section 5, where our RL-based approach is detailed. The results of the numerical evaluation are showcased and discussed in Section 6. Finally, Section 7 concludes the paper with some final remarks.

2. Related Work

Resource management in mmWave access networks has been largely investigated in recent literature, taking into account the new challenges brought in by directional transmissions compared with the conventional omnidirectionality assumption of works on sub-6GHz networks. Several papers have investigated the optimization of bandwidth allocation [6, 18, 27], power allocation [6, 23, 30, 15, 16, 33, 36], beamwidth assignment [30], frame / slots design [28, 15, 34], transmission delay [8, 24, 12].

Among all these works, it draws remarkable attention that a large part of them dedicates to the traffic routing and transmission scheduling problem. Authors in [25] investigate on the performance of different distributed greedy hop-by-hop path selection to the core network. The work in [12] proposes a routing scheme using multiple overlapping spanning trees and schedules transmissions to minimize the end-to-end delay along a subset of paths computed from the routing scheme. Authors in [31] study path selection and rate allocation to maximize the network data rate by leveraging Lyapunov stochastic optimization.

Some works on routing and scheduling take into account the status of links (i.e., line-of-sight (LOS) and non-LOS (NLOS)). The work in [9] performs a slot-by-slot link scheduling to maximize the instantaneous throughput considering the blockage probability in the current slot described according to a discrete-time Markov chain. Authors in [8] present a joint dynamic routing and scheduling policy based on proportional flow delays. Every packet requires a multi-hop path to reach its destination, which is selected on the base of the current network conditions. [11] performs routing and scheduling to improve the end-to-end throughput in the wireless backhaul, targeting at urban environments and utilizing 3D models of buildings as primary blockage sources. [22] maximizes the number of protected flows in a wireless backhaul by selecting relay nodes to bypass the blockages when they occur.

Recent years have also seen a widespread utilization of machine learning techniques, such as reinforcement learning, in mmWave wireless networks. The work in [13] proposes spectrum allocation algorithms based on Double Deep Q-network (DQN) and Actor Critic for IAB networks. [24] proposes a semi-distributed multi-armed bandit learning algorithm to minimize the end-to-end latency in backhaul networks, which is proven to be adaptive to load imbalance, channel variations and link failures. Authors in [31] resort to regret RL to perform route selection and tackle the problem of rate allocation by successive convex approximation method. The work in [30] maximizes data rate by controlling transmitter beamwidth and power by using risk-sensitive RL, while authors in [7] present a DQN based approach to assign backhaul resources to users with blockages. Finally, there have been some successful cases of combination of DRL and LSTM to perform resource management in wireless networks [14, 21] as we do in this article. These works show the good performance of RL methods on dynamic problems with different targets in various network scenarios. However, none of them aims to deal with routing and link scheduling problem in mmWave IAB networks and with the dynamics caused by link blockages.

Despite the significant results that have been achieved in literature, the efforts made in dealing with the dynamic blockage scenario characterizing IAB networks are far from being enough. Some of the existing works assume static blockages for specific propagation scenarios [6] or those caused by urban buildings [11], while others assume simple dynamic blockages [9, 22]. Indeed, these works either use steady state distribution of the link status to compute an expected metric, which cannot capture frame-by-frame or slot-by-slot actual link conditions [9], or make decisions based on the current blockage situation [22], which only aims to maximize the instantaneous throughput of the current time slot, without considering the impact on future time slots.

2.1. Our contribution

Differently from existing works, this article aims to tackle the problem of joint routing and scheduling in mmWave IAB networks, with an emphasis on dealing with dynamic blockages described by a realistic blockage model in rapidly changing environments. Extending our previous work [35] that considers a simplified blockage model relying on a Bernoulli process, we have completely redesigned the neural network (NN) framework to deal with the temporal correlation of the measure-based blockage model developed in [19]. The main contributions of this article are summarized as follows.

- We propose a hybrid approach where optimization and RL techniques jointly contribute to provide an optimal and adaptive solution to the complicated problem of routing and link scheduling in mmWave IAB networks.
- We consider a very realistic scenario, and specifically design an approach tailored to it, in which random blockages remarkably impact on the resource allocation in mmWave IAB networks.
- We provide a CG-based formulation of the problem that can be used both as a reference benchmark for optimality and as a support for our DRL framework.
- We develop a DRL framework based on LSTM to capture the regularity of environment dynamics so as to better adapt to the changing conditions. Its performance advantages with respect to traditional optimization approaches clearly emerge from a numerical evaluation.
- We address implementation and feasibility issues of the proposed DRL framework.
- We implement both offline and online DRL approaches to discuss strengths and weaknesses of an online NN parameters' update rather than only an initial offline NN training.

3. System Overview

In this article, we consider a mmWave 5G access network featuring an IAB architecture. It consists of a multi-hop mmWave wireless network with a gNodeB (IAB-donor), which is directly connected to the core network via a high capacity link (i.e., fiber), a set of self-backhauled IAB-nodes that act as relay nodes for the user traffic to / from the IAB-donor, and user equipment (UE) that can reach IAB-donor either via direct links or through multi-hop IAB-node paths. The *backhaul links* between IAB-donor and IAB-node, or IAB-node and IAB-node, and the *access links* between IAB-donor and UEs, or IAB-nodes and UEs, are wireless and share the same frequency bands (i.e., in-band backhaul), i.e., they interfere with each other. Moreover, we consider here a static 5G Enhanced Mobile Broadband (eMBB) use case (like domestic broadband access, high-throughput gates, digital kiosks, etc.) where UEs can be assumed to be static nodes. This is expected to be the first application of mmWave IAB networks.

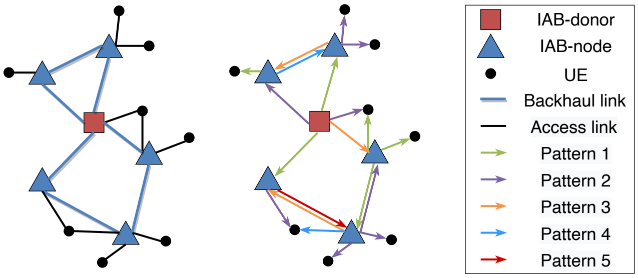


Figure 1: A toy example of IAB network scenario and five patterns constructed by access and backhaul links.

We focus on the downlink traffic flow as it is expected to play the most important role in such networks, leaving the investigation on the uplink transmission to a future work. The main performance figures we will analyze are the average UE throughput, in terms of number of data bits transferred from the IAB-donor to UEs, and the service coverage, which we express as the percentage of UEs in the service area experimenting a non-null throughput.

The network can be represented as a directed graph $\mathcal{G}(\mathcal{V}, \mathcal{L})$, where $\mathcal{V}$ denotes the index set of nodes including IAB-donor, IAB-nodes and UEs, and $\mathcal{L}$ includes all the potential links connecting the nodes in $\mathcal{V}$. If not specified, IAB-donor is deemed as a special IAB-node. Hence, the node set $\mathcal{V}$ is divided into two subsets, namely, IAB-node set $\mathcal{R} \subset \mathcal{V}$ and UE set $\mathcal{U} \subset \mathcal{V}$. Without loss of generality, IAB-donor is regarded as the 0-th IAB-node and the remaining subset of IAB-nodes is represented via their indices $\mathcal{R}_{sub} = 1, 2, \dots, |\mathcal{R}| - 1$.

MmWave 5G access networks are based on a time-division multiplexing / time-division multiple access (TDM / TDMA) resource sharing where each frame involves $T \in \mathbb{N}$ slots with equal duration δ in a time domain $\mathcal{T}$. IAB-nodes transmit in these slots achieving a rate that depends on the signal-to-interference-plus-noise ratio (SINR) available at receivers, thus an interference coordination approach must be adopted not only to activate a sequence of backhaul links to transport traffic through IAB-nodes, but also to schedule access links in the same frame backhaul links are scheduled. In accordance with this premise, IAB networks implement a space-division multiplexing (SDM) approach on top of TDM / TDMA to take advantage of the high directivity of mmWave antennas, allowing multiple concurrent transmissions in each slot. This results in a frame composed of a slot-by-slot sequence of sets of links simultaneously activated, i.e., a sequence of *link patterns*, that satisfy channel conditions (e.g., interference requirements, antenna patterns), half-duplex constraints, multi-beam features, power limits, etc. We will describe them in detail in Section 4. Figure 1 depicts an example of a network scenario with five possible link patterns. Note that a link pattern can include both access and backhaul links. The optimal sequence of activated link patterns allows to maximize the number of data bits transferred from the IAB-donor to UEs, which is equivalent to maximize the downlink throughput of the IAB network.

Frequent obstacle obstructions strongly characterize mmWave links, therefore a dynamic and realistic blockage model has been introduced into our system. When a link is blocked by an obstacle, no data can be transferred. However, IAB-nodes are expected to be installed at relatively high places (e.g., lamp posts, roof tops, etc.) to improve visibility and avoid tampering, therefore we can expect few obstacles to exist there, hence it is less likely for backhaul links to be blocked. In contrast, access links are exposed to more recurrent blockages caused by nomadic obstacles, like those produced by pedestrian and transportation traffic. Based on this observation, we realistically apply blockages only to access links[1]. In particular, we adopt the measurement-based signal fading model analyzed in [19], which characterizes various blockages caused by pedestrian crowds in New York to provide a stochastic blocked duration of a link. A binary semi-Markov link-blockage model is proposed in that work where the blocked duration of each access link follows a *log-normal* distribution. Motivated by the Poisson nature of obstacle blockage events, the non-blocked (available) duration of a link follows a *negative exponential* distribution. The considered probability density functions are shown in (1) where μ and σ are respectively mean and standard deviation of the blocked-link time duration and λ is the obstacle blockage event arrival rate. To coordinate with a time-slotted system, the time spans of both blocked and not-blocked phases (t_B and t_{NB}) are rounded up and expressed in number of slots.

$$\begin{aligned} f(t_B) &= \frac{1}{t_B\sigma\sqrt{2\pi}}e^{-(\ln t_B-\mu)^2/2\sigma^2} \\ f(t_{NB}) &= \lambda e^{-\lambda t_{NB}} \end{aligned} \tag{1}$$

Resource allocation in wireless networks has been traditionally carried out via optimization-based approaches, which can provide optimal solutions in quasi-static scenarios. When facing the dynamic environment of mmWave access networks (e.g., frequent topology changes due to recurring link blockages), it is infeasible to resort to optimization methods to compute a near-optimal scheduling scheme on-the-fly, due to their time-consuming algorithms. To tackle the complicated blockage scenario described above, RL can be the ideal solution to capture the intrinsic regularities of the dynamic environment and learn how to provide a robust network schedule well performing in realistic scenarios. However, RL can get stuck into local optima. Therefore, we present next our hybrid approach that joins together advantages of both optimization and RL techniques.

Although indirectly, we believe our approach can play a part in improving the network energy efficiency. Indeed, (1) the generation of optimized link patterns to transfer a large number of bits in each slot requires to reduce the impact of interference, thus to minimize the transmission power for every established link; (2) the ability of our RL approach to learn of potentially obstructed links allows not to waste energy on transmitting data that will not be correctly decoded at the receiver due to the blockage.

Table 1
Summary of notations in Section 4.1

Parameter	Definition
$\mathcal{V}$, $\mathcal{L}$	Set of nodes and set of potential links
$\mathcal{R}$, $\mathcal{U}$	Set of IAB-nodes and set of UEs
$\mathcal{E}$, $\mathcal{E}_R$	Pattern set and restricted pattern set
$\mathcal{E}_{CG}$	Set of patterns obtained from CG pricing
T, δ	Number of slots in a frame and the length of a slot
$B^e_{BH(i,j)}$	Amount of bits delivered over backhaul link (i,j) in pattern e
$B^e_{ACC(i,u)}$	Amount of bits delivered over access link (i,u) in pattern e
ρ_{min}	Minimum data rate required for each user
C_{donor}	Capacity value used to assign IAB-donor role in IAB-nodes
Variable	Definition
λ_e	Number of slots in which pattern e is activated
$f^{BH}_{i,j,k}$	Flow bits from i to j directed to k $(i,j,k \in \mathcal{R})$
$f^{ACC}_{i,u}$	Flow bits from $i \in \mathcal{R}$ to UE $u \in \mathcal{U}$
$f^{BH}_{CN,r,h}$	Flow bits from core to IAB-donor r directed to $h \in \mathcal{R}$

4. Optimization Approach to Resource Allocation

From an optimization standpoint, the problem of maximizing the performance of a mmWave IAB network can be reduced to a variant of the traditional problem of resource optimization in wireless multi-hop networks [5], in which flow routing and link scheduling are jointly optimized while ensuring fairness among UEs and meeting mmWave-specific physical requirements, such as half-duplex, simultaneous multiple beams, directional interference, and transmission power limitations.

4.1. Pattern-based Formulation

Link-based decision variables are usually considered in optimization problem formulations, often leading to intractable mathematical programs that can be solved in reasonable time only for instances of very limited size. To overcome this limitation, a different set of decision variables can be defined. The idea of *link patterns* introduced in Section 3 can be leveraged and decision variables can be set to represent those compatible sets of links that can be simultaneously activated meeting both hardware specifications and SINR requirements. We denote with $\mathcal{E}$ the set of all the potential patterns, and associate an integer decision variable λ_e to each pattern $e \in \mathcal{E}$. The value of λ_e denotes the number of slots in which pattern e is activated. Considering that only one pattern can be activated per time slot, the overall number of times all patterns are activated provides the number of occupied slots. Resorting to pattern-based decision variables also has the advantage to allow to separate the routing and scheduling problem from the link coexistence problem originated from physical constraints. Indeed, routing and scheduling needs only to be fed up with feasible patterns, regardless the assumptions and the procedures patterns are

[1] Note that this is not a limitation of our scenario, but rather an effort to make it more realistic. Indeed, backhaul link blockages can be straightforwardly included in the approach if the specific use case needs it.

generated with.

We define flow variable $f^{BH}_{i,j,k}$ as the total number of bits in a frame of the flow from IAB-node i to IAB-node j directed to IAB-node k (hence $f^{BH}_{i,j,j}$ denotes the flow directly from IAB-node i to IAB-node j) and $f^{ACC}_{i,u}$ as the number of bits in a frame sent over the access link between IAB-node i and UE u. Then, the mixed-integer linear programming (MILP) formulation for the resource allocation problem is given by:

$$\max \sum_{r\in\mathcal{R},u\in\mathcal{U}} f^{ACC}_{r,u}, \tag{2a}$$

$$s.t. \sum_{r\in\mathcal{R}} f^{ACC}_{r,u} \geqslant \rho_{min} T\delta,\ \forall u \in \mathcal{U}, \tag{2b}$$

$$\sum_{\substack{n\in\mathcal{R}:\\ n\neq r}} f^{BH}_{n,r,r} + f^{BH}_{CN,r,r} = \sum_{u\in\mathcal{U}} f^{ACC}_{r,u}, \quad \forall r \in \mathcal{R}, \tag{2c}$$

$$\sum_{\substack{n\in\mathcal{R}:\\ n\neq r}} f^{BH}_{n,r,h} + f^{BH}_{CN,r,h} = \sum_{\substack{m\in\mathcal{R}:\\ m\neq r}} f^{BH}_{r,m,h}, \quad \forall r,h \in \mathcal{R} : r \neq h, \tag{2d}$$

$$\sum_{h\in\mathcal{R}} f^{BH}_{CN,r,h} \leq \begin{cases} C_{donor} & r = 0 \\ 0 & r \neq 0 \end{cases},\ \forall r \in \mathcal{R}, \tag{2e}$$

$$\sum_{e\in\mathcal{E}} B^{e}_{BH(i,j)} \lambda_e \geq \sum_{h\in\mathcal{R}} f^{BH}_{i,j,h}, \quad \forall i,j \in \mathcal{R} : i \neq j, \tag{2f}$$

$$\sum_{e\in\mathcal{E}} B^{e}_{ACC(i,u)} \lambda_e \geq f^{ACC}_{i,u}, \quad \forall i \in \mathcal{R}, u \in \mathcal{U}, \tag{2g}$$

$$\sum_{e\in\mathcal{E}} \lambda_e \leq T, \tag{2h}$$

$$f^{BH}_{CN,i,k}, f^{BH}_{i,j,k}, f^{ACC}_{i,u} \in \mathbb{R}^{+},\ \forall i,j,k \in \mathcal{R}, u \in \mathcal{U}, \tag{2i}$$

$$\lambda_e \in \mathbb{Z}^{+}, \quad \forall e \in \mathcal{E}. \tag{2j}$$

We refer to Table 1 for the complete list of parameters and variables.

Routing constraints (2c)-(2e) Traffic originates from the core network and flows to UEs via the IAB-donor and the IAB-nodes. We denote as $f^{BH}_{CN,r,h}$ the core-network flow entering IAB-donor and directed to IAB-node h. A core-network flow is allowed only at node with a non-null capacity C_{donor}. If node r is an IAB-donor, namely $r = 0$, $f^{BH}_{CN,r,h} > 0$; otherwise, $f^{BH}_{CN,r,h} = 0$. This is enforced by constraint (2e).

Flow balance equation (2c) states that the incoming flow to a destination IAB-node r from both the other IAB-nodes and the core network must equalize the flow sent to all the UEs served by r. Similarly, enforced by (2d), the incoming flow to an intermediate IAB-node r along the path to IAB-node h must equalize to the outgoing flow directed to h.

Scheduling constraints (2f)-(2h) The flows defined by the previous set of constraints must be supported by the scheduling of proper link patterns. The activation of link pattern e provides a number of bits to be transmitted over each link in the single slot given by parameters $B^{e}_{BH(i,j)}$ and $B^{e}_{ACC(i,u)}$ for, respectively, backhaul and access links. The sequence of activated link patterns defines a total number of bits in a frame transmitted along each link, and this number must be larger than or equal to the one indicated by flow variables to obtain a consistent solution. This aspect is enforced by constraint (2f) for backhaul links and (2g) for access links. Finally, constraint (2h) states that the number of activated pattern e, each considered with the multiplicity λ_e, must not exceed the frame length T.

Optimization objective (2a)-(2b) Since mmWave access networks are envisioned to provide very high throughput to the users in the service area, we jointly optimize routing and scheduling to maximize the total flow of bits received by UEs in a frame, as indicated in the objective function (2a). However, a mere throughput maximization often leads to solutions that prioritize some well-positioned UEs, excluding many others from the service. In order to avoid such an undesirable behavior, we set a min-rate constraint (2b) to guarantee fairness among users, where ρ_{min} is the minimum required data rate that each user has to achieve and δ is the slot temporal duration.

Optimally solving the pattern-based problem formulation presented above requires us to provide as input the whole set of possible patterns that can be activated. However, this set has a cardinality that increases exponentially with the number of links, thus creating a formulation with a huge number of variables, still making the problem potentially intractable. In order to solve this issue, the Column Generation (CG) technique can be applied.

In CG, only a subset of λ_e variables is considered at the beginning. Denoting the formulation (2) as *Master Problem (MP)*, we refer to *Restricted Master Problem (RMP)* to indicate formulation (2) with a restricted set of pattern variables $\lambda_e : e \in \mathcal{E}_R \subset \mathcal{E}$, where $\mathcal{E}_R$ is a restricted pattern set. The solution of the RMP provides a selection of link patterns, namely a frame, that is optimal for RMP, but may be not for the original MP, as only a subset of variables (patterns) is considered. We need a procedure, namely Column Generation, to check whether the solution obtained is also optimal for the original MP or to find out the variables (patterns) to be included into the pattern set to further improve the solution. Every time a new objective-improving pattern is found, it is added to the set of available patterns, and the process iterates until no improving pattern can be generated. At the end of the CG process, the final obtained pattern set $\mathcal{E}_{CG}$ provides a good set of candidates to solve MP. The practice also shows $\mathcal{E}_{CG}$ to be of limited size with respect to the set of all possible link patterns.

The set $\mathcal{E}_{CG}$ includes link patterns built considering nodes' hardware constraints, SINR thresholds, channel and antenna gains, power levels, etc., under the assumption that links are always available, thus no random obstacle is considered. From a point of view, this is a limitation, as scheduled patterns can incur link blockages, limiting the expected bit transfer of the current and next slots. However, set $\mathcal{E}_{CG}$ includes the best mix of link activations to support the optimal

Table 2
Summary of notations in Section 4.2

Parameter	Definition
$\mathcal{L}^{BH}, \mathcal{L}^{ACC}$	Sets of backhaul and access links
$\mathcal{M}^{BH}, \mathcal{M}^{ACC}$	Sets of MCSs for backhaul and access
R_m^{BH}, R_m^{ACC}	Bitrates with MCS m for backhaul and access
$\gamma_m^{BH}, \gamma_m^{ACC}$	SINR thresholds of MCS m for backhaul and access
G^{BBBB}, G^{BBBA}	Channel gain from backhaul to backhaul and access
G^{BABB}, G^{BABA}	Channel gain from access to backhaul and access
η_{IAB}, η_{UE}	Noises at IAB-node and UE receiver
COV^{BH}	Coverage matrix of backhaul
COV^{ACC}	Coverage matrix of access
$\mathcal{P}_i$	Set of panels of IAB-node i
$K_{i,p}^{TX}$	Multi-beams at panel p of node i for transmission
$K_{i,p}^{RX}$	Multi-beams at panel p of node i for reception
$TxP_{i,j}^{BH}$	Panel ID of node i used on backhaul link (i, j)
$TxP_{i,u}^{ACC}$	Panel ID of node i used on access link (i, u)
$RxP_{i,j}^{BH}$	Panel ID of node j from which node i receives
$P_{i,p}^{MAX}$	Total power available at panel p of node i
$P_{i,p}^{MIN}$	Minimum activation power at panel p of node i
Variable	Definition
$\beta_{i,j},\ \alpha_{i,u},\ \psi$	Dual variables w.r.t. constraints 2f, 2g, 2h
$z_{i,j,m}^{BH}$	Whether to activate backhaul link (i, j) with MCS m
$w_{i,u,m}^{ACC}$	Whether to activate access link (i, u) with MCS m
p^{BH}, p^{ACC}	Transmission power allocated to backhaul and access
b_i^{RX}	Whether IAB-node i is in reception
$b_{i,p}^{p-TX}$	Whether panel p of IAB-node i is in transmission
b_i^{TX}	Whether at least a panel of node i is in transmission

routing and scheduling enforced by MP. Therefore, among all possible link patterns, $\mathcal{E}_{CG}$ forms the most promising set of actions that an RL agent can play to achieve a good performance. Note that set $\mathcal{E}_{CG}$ depends only on hardware configurations and nodes locations, thus can be computed a priori once the IAB network is deployed: it is not an output of the RL process, but rather an input that defines its action space and must be computed once for all.

Given the above considerations, we can use the *CG approach to achieve a twofold result*:

1. Solving MP using formulation (2) with the link-pattern set $\mathcal{E}_{CG}$ to obtain a quasi-optimal solution of instances of any size, whose results will be the main *benchmark* to be compared against those achievable with the RL approach we propose in this article.
2. Exploiting CG to be a *link-pattern generator* that creates the action space of our RL approach, in which we will learn the best sequence of link-patterns (among those in $\mathcal{E}_{CG}$) to apply in a dynamic environment to pursue a fair user throughput maximization as in MP.

4.2. Link Pattern Generation

The generation of new promising link patterns in the CG approach is based on the assessment of the new pattern's potential to improve the MP's objective function. This procedure is called *pricing* and relies on the continuous relaxation of the original MP that we name as *lin-MP*, and in particular, on the dual variables of lin-MP.

We recall that, given a solution of a primal problem lin-MP, if the dual variables related to such solution are feasible for lin-MP's dual problem, then the given primal solution is optimal for lin-MP. Besides, each variable (constraint) of lin-MP is associated to a constraint (variable) of its dual problem. Given a primal variable, if the associated dual constraint is violated, the considered variable has a positive reduced cost and therefore can produce an improvement in the objective function if it is added to the set of the considered variables. Indeed, this inclusion can potentially provide a positive contribution to the object of the maximization. Therefore, the aim of the pricing procedure is to generate a new feasible pattern (a new variable) with a positive reduced cost such that the related dual constraint is violated. The variable associated to this pattern must then be added to lin-MP, the pattern included in the link-pattern set, and lin-MP solved again. Then, the generation repeats. The generation stops when the optimal solution of lin-MP is achieved, that is, when no pattern can be built such that the related dual constraint is violated.

Consider formulation (2). Denoting with $\beta_{i,j}$ the dual variable related to constraint (2f), $\alpha_{i,u}$ the dual variable related to constraint (2g), and ψ the dual variable related to constraint (2h), the dual constraint associated to a given pattern e is

$$\sum_{(i,j)\in\mathcal{L}^{BH}} B^{e}_{BH(i,j)}\beta_{i,j} + \sum_{(i,u)\in\mathcal{L}^{ACC}} B^{e}_{ACC(i,u)}\alpha_{i,u} - \psi \le 0, \quad (3)$$

where $\mathcal{L}^{BH} = \{(i, j) \in \mathcal{R} \times \mathcal{R} : i \neq j\}$ and $\mathcal{L}^{ACC} = \{(i, u) \in \mathcal{R} \times \mathcal{U}\}$ are the sets of wireless backhaul links and access links, respectively. Note that we have considered only dual variables associated to constraints where primal pattern variable λ_e does appear, as this is the primal variable we need to generate. We refer to Table 2 for the complete list of parameters and variables.

To solve the pricing problem we must look for a new pattern e. We express with binary variables $z^{BH}_{i,j,m}$ and $w^{ACC}_{i,u,m}$ that take value 1 when deciding to activate in pattern e backhaul link (i, j) and access link (i, u) with Modulation and Coding Scheme (MCS) m, respectively. The new pattern must satisfy feasibility constraints (namely, hardware constraints and SINR thresholds described in Section 4.2.1) and, in order to find a new pattern that violates the dual constraint (3), we must maximize the following quantity:

$$\sum_{\substack{(i,j)\in\mathcal{L}^{BH},\\ m\in\mathcal{M}^{BH}}} R^{BH}_m \delta\beta_{i,j}\, z^{BH}_{i,j,m} + \sum_{\substack{(i,u)\in\mathcal{L}^{ACC},\\ m\in\mathcal{M}^{ACC}}} R^{ACC}_m \delta\alpha_{i,u}\, w^{ACC}_{i,u,m} - \psi, \quad (4)$$

where $\mathcal{M}^{ACC}$ and $\mathcal{M}^{BH}$ are, respectively, the set of MCSs available for access and backhaul links, and R^{ACC}_m and R^{BH}_m are the bitrates achievable with the m-th MCS in the set of those available for access and backhaul links, respectively.

The pricing guarantees that, if a solution with a positive objective function value is found, the dual constraint is violated and the pattern must be added to the set of those available. Thus, we should add a new pattern e_n to $\mathcal{E}$ by setting

$$B^{e_n}_{BH(i,j)} = \sum_{m\in\mathcal{M}^{BH}} R^{BH}_m \delta\, z^{BH}_{i,j,m} \quad \forall (i, j) \in \mathcal{L}^{BH}, \quad (5)$$

$$B^{e_n}_{ACC(i,u)} = \sum_{m\in\mathcal{M}^{ACC}} R^{ACC}_m \delta\, w^{ACC}_{i,u,m} \quad \forall (i, u) \in \mathcal{L}^{ACC}. \quad (6)$$

Vice versa, if a negative solution or no solution is found, we can certify that no other patterns would improve the objective function value. Therefore, the CG process can stop.

Alternatively, we can put a maximum number of CG iterations to arbitrarily approximate the optimal set of patterns.

As a final step, when an optimal set of available patterns $\mathcal{E}_{CG}$ has been found for lin-MP, the original problem MP can be solved removing the continuous relaxation and considering $\mathcal{E} = \mathcal{E}_{CG}$. This final integer solution will provide a heuristic result. Indeed, $\mathcal{E}_{CG}$ is not guaranteed to be optimal for the original problem MP as the generated pattern set is optimal just for lin-MP, the continuous relaxation, and may not be so for the integer version. However, many works in literature [4, 5, 17] show that results very close to the optimum can be achieved and a small gap can be obtained in very short execution time. We present now the constraints we must consider to provide a link pattern that is feasible from a technological point of view.

4.2.1. *Link pattern constraints*

During the CG process, we generate admissible patterns by considering all the technological and practical aspects arising when we activate simultaneous links in a mmWave IAB scenario, such as the channel model, the SINR values required to activate specific MCSs, the availability of power control to reduce the interference impact, half-duplex constraints, etc. These aspects also include how devices are engineered. Details like the number of antenna panels and the number of beams that can be simultaneously activated by the panel in the pattern must be taken into account. The main sets of constraints necessary to capture these aspects are described in the following.

SINR constraints SINR constraints are fundamental to implement a realistic MCS selection when a link is activated. We define two SINR thresholds γ_m^{ACC} and γ_m^{BH} to indicate the SINR values necessary to activate MCS m over an access and a backhaul link, respectively. We compute received powers using a channel gain matrix model, in which the transmission power of node i is multiplied by the channel gain matrix's element (i, j) to provide the received power at node j. Channel gain includes not only path loss but also hardware- and antenna-related gain value. Since mmWave transmissions are highly directional, the antenna's pointing direction is important for computing correct received power and interference and must appear in channel gain matrices.

We introduce four channel gain matrices: $G_{i \to j, r \to n}^{BBBB}$, $G_{i \to u, r \to n}^{BABB}$, $G_{i \to j, r \to q}^{BBBA}$ and $G_{i \to u, r \to q}^{BABA}$. The first two matrices express the channel gains between two IAB-nodes. $G_{i \to j, r \to n}^{BBBB}$ represents the channel gain between transmitter i and receiver n when i transmits to IAB-node j and n receives from IAB-node r. The specification of nodes j and r allows to properly compute channel gains, considering antenna directionality and different pointing directions. Similarly, $G_{i \to u, r \to n}^{BABB}$ indicates the channel gain between IAB-node i and IAB-node n when i transmits to UE u and n receives from IAB-node r. The other two matrices express the channel gains between IAB-nodes and UEs. Specifically, $G_{i \to j, r \to q}^{BBBA}$ indicates the channel gain between IAB-node i's transmitter and UE q's receiver when i transmits to IAB-node j and q receives from IAB-node r. $G_{i \to u, r \to q}^{BABA}$ is the channel gain between IAB-node i's transmitter and UE q's receiver when i transmits to UE u and q receives from IAB-node r. Figure 2 illustrates the considered channel gain model. Channel gain matrices, together with the assumption that transmitting nodes point their antennas towards their intended receivers, allow us to provide a complete description of the propagation conditions that the players of the considered scenario can meet. Note that these matrices can be computed before the optimization, relying on both statistical models and measurement campaigns. Also, this matrix-based approach makes the model independent of any fine technological and propagation detail, which can be computed offline and directly included in the matrices.

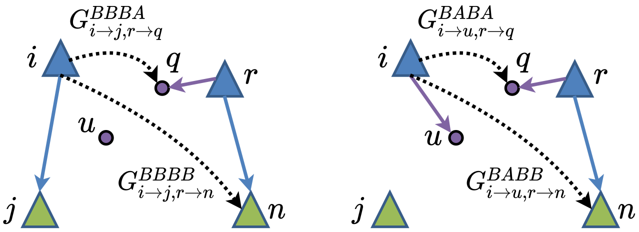

$G_{i \to j, r \to q}^{BBBA}$
i
q
r
u
$G_{i \to j, r \to n}^{BBBB}$
j
n
$G_{i \to u, r \to q}^{BABA}$
i
q
r
u
$G_{i \to u, r \to n}^{BABB}$
j
n

Figure 2: Channel gain model. Blue and green triangles represent the transmitter and receiver IAB nodes which are connected with backhaul links shown as blue arrows. Purple dots are the UEs served with access links marked by purple arrows. The dashed black arrows indicate the channel gains.

Based on the above-mentioned channel gains, SINR constraints for backhaul links and access links can be written as in Equation (7a), for IAB-node receivers, and Equation (7b), for UEs. The left-hand side of each equation expresses the power available at receivers of the link (from IAB-node i to IAB-node j or from IAB-node i to UE u), where $p_{i,j}^{BH}$ and $p_{i,u}^{ACC}$ respectively refer to the transmission power from IAB-node i to IAB-node j or to UE u. The right-hand side includes the total interference power from other simultaneously active links and the noises at IAB-node and UE receiver, respectively, η_{IAB} and η_{UE}. SINR conditions must be satisfied considering thresholds γ_m^{BH} and γ_m^{ACC} associated to the MCS m. Note that the $M(\cdot)$ term derives from a Big-M linearization technique that deactivates SINR constraints over links that are not selected to be active with the MCS m (or not active at all), namely $z_{i,j,m}^{BH} = 0$ and $w_{i,u,m}^{ACC} = 0$.

Coverage constraints IAB-node i can transmit to UE u or to another IAB-node j only if node i can cover u or reach j. The coverage is evaluated by checking whether the lowest access or backhaul MCS can be achieved over the link activated in isolation. If such an MCS cannot be achieved, the corresponding elements in the binary matrices $COV_{i,u}^{ACC}$ and $COV_{i,j}^{BH}$ are set to 0. They take value 1 otherwise. The constraints are:

$$z_{i,j,m}^{BH} \leq COV_{i,j}^{BH}, \quad \forall (i,j) \in \mathcal{L}^{BH}, m \in \mathcal{M}^{BH}, \tag{8a}$$

$$w_{i,u,m}^{ACC} \leq COV_{i,u}^{ACC}, \quad \forall (i,u) \in \mathcal{L}^{ACC}, m \in \mathcal{M}^{ACC}. \tag{8b}$$

Note that these constraints are not strictly required, as previous SINR constraints prevent out-of-coverage links from be-

$$p_{i,j}^{BH} G_{i\to j,i\to j}^{BBBB} + M\left(1 - z_{i,j,m}^{BH}\right) \geq \gamma_m^{BH}\left(\eta_{IAB} + \sum_{\substack{h,k\in\mathcal{R}:\\ k\neq j}} p_{h,k}^{BH} G_{h\to k,i\to j}^{BBBB} + \sum_{h\in\mathcal{R},u\in\mathcal{U}} p_{h,u}^{ACC} G_{h\to u,i\to j}^{BABB}\right), \quad \forall (i,j)\in\mathcal{L}^{BH}, m\in\mathcal{M}^{\mathcal{BH}} \tag{7a}$$

$$p_{i,u}^{ACC} G_{i\to u,i\to u}^{BABA} + M\left(1 - w_{i,u,m}^{ACC}\right) \geq \gamma_m^{ACC}\left(\eta_{UE} + \sum_{h,k\in\mathcal{R}} p_{h,k}^{BH} G_{h\to k,i\to u}^{BBBA} + \sum_{\substack{h\in\mathcal{R},q\in\mathcal{U}:\\ q\neq u}} p_{h,q}^{ACC} G_{h\to q,i\to u}^{BABA}\right), \quad \forall (i,u)\in\mathcal{L}^{ACC}, m\in\mathcal{M}^{ACC}. \tag{7b}$$

ing activated. However, they remarkably speed up the solution process by efficiently cutting unfeasible solutions from the exploration tree of the optimization solver.

Half-duplex and multiple beams constraints The most common realization of an IAB-node consists of a set of 3 or 4 panels mounted on an node according to a triangle or a square form factor to cover the entire node's surrounding. Given a node i, each panel p in the set of node's panels, $\mathcal{P}_i$, is equipped with a number of RF chains that allow the panel to transmit up to $K_{i,p}^{TX}$ or to receive up to $K_{i,p}^{RX}$ simultaneous streams (beams) to / from neighboring nodes. If at least one link in a panel of node i is active in reception, the node i is declared active in reception and the binary decision variable b_i^{RX} is set to 1, as stated by constraint (9a). Due to the power allocation constraints introduced later, we follow a slightly different approach for transmissions. If at least one link in panel p of node i is active in transmission, the panel is declared active in transmission and the binary decision variable $b_{i,p}^{p-TX}$ is set to 1. Similarly, if at least one panel is active in transmission, the entire node i is declared active in transmission and the binary decision variable b_i^{TX} is set to 1. This behavior is enforced by constraints (9b) and (9c).

The binary variables allow to enforce an IAB-node to operate in a half-duplex mode, that is, it cannot be active both in transmission and in reception in the same slot[2], as ensured by constraint (9d). Finally, according to the current hardware specifications, we assume that UEs can receive from at most one IAB-node in each pattern (slot), as stated by constraint (9e). The constraints are defined as follows:

$$\sum_{\substack{j\in\mathcal{R},m\in\mathcal{M}:\\ RxP_{i,j}^{BH}=p}} z_{j,i,m}^{BH} \leq K_{i,p}^{RX} b_i^{RX}, \quad \forall i\in\mathcal{R}, p\in\mathcal{P}_i, \tag{9a}$$

$$\sum_{\substack{m\in\mathcal{M},j\in\mathcal{R}:\\ TxP_{i,j}^{BH}=p}} z_{i,j,m}^{BH} + \sum_{\substack{m\in\mathcal{M},u\in\mathcal{U}:\\ TxP_{i,u}^{ACC}=p}} w_{i,u,m}^{ACC} \leq b_{i,p}^{p-TX} K_{i,p}^{TX}, \quad \forall i\in\mathcal{R}, p\in\mathcal{P}_i, \tag{9b}$$

$$b_{i,p}^{p-TX} \leq b_i^{TX}, \quad \forall i\in\mathcal{R}, p\in\mathcal{P}_i, \tag{9c}$$

$$b_i^{RX} + b_i^{TX} \leq 1, \quad \forall i\in\mathcal{R}, \tag{9d}$$

$$\sum_{j\in\mathcal{R}} w_{j,u}^{ACC} \leq 1, \quad \forall u\in\mathcal{U}, \tag{9e}$$

where $TxP_{i,j}^{BH}\in\mathcal{P}_i$ and $TxP_{i,u}^{ACC}\in\mathcal{P}_i$ are input parameters that respectively provide the ID of node i's panel to be used for the transmission to IAB-node j and UE u, while $RxP_{i,j}^{BH}\in\mathcal{P}_i$ defines the panel from which node i receives node j's transmissions. Note that these parameters can be computed a priori as they depend on the nodes' placement.

Power allocation constraints We assume that the power assigned to each IAB-node beam can change at every slot to follow SINR constraints and a per-panel power budget can be applied on IAB nodes[3]. Constraints (10a) and (10b) ensure that no power can be allocated to a transmission if the associated link is not activated in the pattern.

We capture a common feature of real hardware for which the panel transmission power can be tuned only between a minimum and maximum value. Constraint (10c) enforces that the overall power allocated for concurrent transmissions at the panel p of node i must not exceed the total power available at that panel, $P_{i,p}^{MAX}$. Constraint (10d) imposes a minimum activation power $P_{i,p}^{MIN}$ to panel p of node i, which must be shared among simultaneously activated links.

$$p_{i,j}^{BH} \leq P_{i,TxP_{i,j}^{BH}}^{MAX} \sum_{m\in\mathcal{M}} z_{i,j,m}^{BH}, \quad \forall (i,j)\in\mathcal{L}^{BH}, \tag{10a}$$

$$p_{i,u}^{ACC} \leq P_{i,TxP_{i,u}^{ACC}}^{MAX} \sum_{m\in\mathcal{M}} w_{i,u,m}^{ACC}, \quad \forall (i,u)\in\mathcal{L}^{ACC}, \tag{10b}$$

$$\sum_{\substack{j\in\mathcal{R}:\\ TxP_{i,j}^{BH}=p}} p_{i,j}^{BH} + \sum_{\substack{u\in\mathcal{U}:\\ TxP_{i,u}^{ACC}=p}} p_{i,u}^{ACC} \leq P_{i,p}^{MAX}, \quad \forall i\in\mathcal{R}, p\in\mathcal{P}_i, \tag{10c}$$

$$\sum_{\substack{j\in\mathcal{R}:\\ TxP_{i,j}^{BH}=p}} p_{i,j}^{BH} + \sum_{\substack{u\in\mathcal{U}:\\ TxP_{i,u}^{ACC}=p}} p_{i,u}^{ACC} \geq P_{i,p}^{MIN} b_{i,p}^{p-TX}, \forall i\in\mathcal{R}, p\in\mathcal{P}_i. \tag{10d}$$

5. Adaptive Resource Allocation

In this section, the basic aspects of deep reinforcement learning (DRL) and recurrent neural network (RNN) are introduced. Subsequently, the flow routing and link scheduling problem is reformulated as a buckets-pipes game. Then, based on the patterns provided by the CG method, a DRL-based approach is presented in detail. To adapt the DRL model to the dynamic IAB scenario, an online framework is presented at last and feasibility issues are discussed.

[2] Note that this is a hardware constraint that can be easily removed when the considered devices can work full-duplex by adopting cancellation techniques to maintain a sufficient level of isolation between transmitting and receiving panels.

[3] This is a realistic assumption based on current prototype device specifications. However, other types of power budget model (like per-node or per-beam) can be easily included in the formulation with small modifications of the power allocation constraints.

5.1. Deep Reinforcement Learning and Recurrent Neural Networks

Reinforcement learning (RL) is widely adopted to optimize decision making and control via sequential interactions between an agent and the environment through accumulated experience following a trial-and-error strategy. Specifically, at time step t, the environment is in the state S_t, and, conditional on this state, the agent selects an action A_t according to the current policy π and then executes it in the environment. At time step $t+1$, based on its reaction to A_t, the environment transits to the state S_{t+1} with some probability and gives a reward R_t back to the agent. The agent adjusts the policy π so as to maximize the long-term cumulative reward, namely the *expected return* $\mathbb{E}_\pi[G_t] = \mathbb{E}[\sum_{k=t+1}^{T} \gamma^{k-t-1} R_k]$. T is the terminal step in an episode, a basic sequence of interactions between the agent and the environment, and γ is a discount factor controlling the importance of a future reward to the current utility. This maximization can be achieved by two categories of algorithms, namely based on value functions and based on policy gradient.

Value-function approaches pick actions at each state in accordance with values estimated via value functions: either state-value functions or action-value functions. A state-value function $v_\pi(S_t) = \mathbb{E}_\pi[G_t|S_t]$ is the expected return from state S_t, following the policy π. An action-value function $q_\pi(S_t, A_t) = \mathbb{E}_\pi[G_t|S_t, A_t]$ is the expected return from state S_t, taking action A_t and following policy π afterwards. The best policy π^* is the strategy that instructs the agent to take the action that leads to the largest value, thus expected future reward, at each state. The value functions can be expressed in tabular form or being approximated as a function of states and actions, such as linear functions, kernels or deep neural networks (DNNs). DRL employs DNNs to approximate value functions.

Policy gradient approaches directly represent policy as a probability function of taking an action at a given state depending on the parameter vector θ, or in other words, $\pi(A_t|S_t;\theta) = Pr\{A_t|S_t;\theta\}$. Like value functions, $\pi(A_t|S_t;\theta)$ can also be represented by a DNN where θ stores the connection weights. θ is updated by applying approximate gradient ascent to $\nabla_\theta \mathbb{E}[R_t]$, whose unbiased estimate is $\nabla_\theta \log \pi(A_t|S_t;\theta)R_t$. To reduce the variance, a baseline is often subtracted from the unbiased estimate. A common choice is to use the estimated state-value function $v(S_t)$ as baseline: $\nabla_\theta \log \pi(A_t|S_t;\theta)(R_t - v(S_t))$. In this formula, $\pi(A_t|S_t;\theta)$ behaves as an actor, indicating to the RL agent the action to perform, and $v(S_t)$ as a critic, providing a quality assessment of the achieved state. Both $v(S_t;\psi)$ and $\pi(A_t|S_t;\theta)$ can be approximated by a NN. The actor-critic technique derives from policy gradient methods, but incorporates the strengths of value functions. Since R_t is an estimate of $q(A_t, S_t)$, the scaling factor $R_t - v(S_t)$ of the policy gradient can represent the advantage of taking action A_t over the other actions at state S_t and be written as $a(A_t, S_t) = q(A_t, S_t) - v(S_t)$. This outlines the definition of advantage actor critic (A2C) approaches [20].

Recurrent neural networks (RNNs) have been widely applied to sequential data in which the time order is strongly relevant, e.g., in natural language processing tasks or stock market predictions. Differently from basic NNs, RNN's output at the current step depends not only on the current input but also on a hidden state that stores the relevant information about the data sequence history, so as to capture time-varying dynamics. However, RNNs only perform well when short-term memory is required, due to the vanishing gradient problem encountered in the back propagation process. To address this issue, LSTM [10] network has been proposed to allow a NN to embrace both short-term and long-term memory by introducing data processing gates (i.e., forget gate, input gate, output gate) to regulate the flow of information in a memory cell. LSTM networks have seen many successful wireless network applications based on variable memory length, sometimes in connection with DRL. The most relevant works are discussed in Section 2.

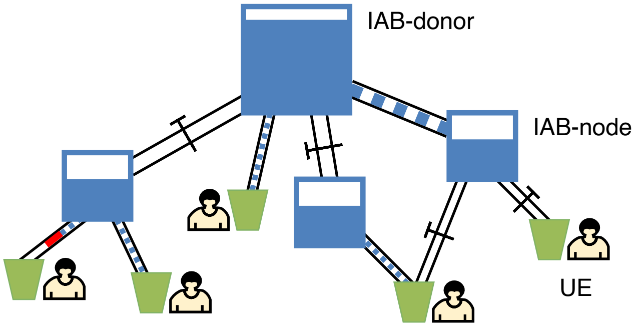


Figure 3: Buckets-pipes game formulation. The blue tanks with water stored represent IAB-donor and IAB nodes, which are connected to UEs' green buckets with pipes. The thick and thin pipes are respectively the equivalents of backhaul and access links with different capacities. A pipe can be clogged with obstacles denoted as the lower-left red patch. The water (data) flows shown in blue dashed lines comply with the rules derived from Section 4.

The ability of LSTM network to deal in a compact way with the history of a data sequence makes it the ideal candidate in building the NNs to approximate actor and critic in the considered scenario. Indeed, since a realistic mmWave link blockage behavior exhibits a strong temporal component, considering the history of this behavior allows the RL agent to select better link patterns to be activated in each slot.

5.2. Buckets-Pipes Game Formulation

In order to perform flow routing and link-pattern scheduling in an RL environment, we reformulate the system model as a game for an RL agent. IAB-donor, IAB-nodes, and UEs are regarded as buckets that store data bits as water. Wireless links act as pipes of different capacities connecting these buckets, which are controlled by valves. The link pattern activated in a slot corresponds to a group of pipes' valves to be opened, letting the water flow through the pipes. Wireless links experimenting blockage situations are equivalent to temporarily clogged pipes. The game's objective is to maximize the total amount of water reaching UEs' buckets in a frame. The buckets-pipes game is illustrated in Figure 3.

Note that in comparison with the optimization approach described in Section 4, the optimization objective (2a) corresponds to the long-term expected return RL aims to maximize. Also, technological constraints (7), (8), (9), and (10) are already satisfied in the sets of candidate actions (patterns) available to the agent. Indeed, it is built from link patterns generated with the CG approach. The overall flow routing, pattern scheduling, and provided fairness depend on which patterns the RL agent selects and how the traffic flows through the system. The RL agent selects one link pattern (pipes' valves) to activate in each slot based on the action probability given by its action policy, while the data transmission obeys the following rules. Traffic is buffered in queues[4] at IAB-nodes, and can be transmitted only to the reachable IAB-nodes and UEs (see coverage constraints in (8)). The total number of bits transferred in a slot from an IAB-node through its outgoing links is limited by the number of bits in its queue, as indicated by flow balance equations (2c) and (2d). Similarly, the maximum number of bits each link can transmit is limited by its capacity, which is determined by the activated pattern, according to constraints (2f) and (2g). UE throughput fairness (2b) is pursued by equally sharing the number of currently buffered data bits transmitted along multiple links originated at each single IAB-node, which appear in the same link pattern.

5.3. Flow Allocation and Pattern Scheduling based on LSTM-Assisted A2C

The buckets-pipes game formulated in the previous section forms the environment the RL agent interacts with. Note that, according to 5G IAB specifications, the IAB-donor is in charge of managing the entire IAB network, therefore we can assume this RL agent to be hosted in the IAB-donor and act as a centralized controller. One slot of the frame is equivalent to one step of RL interaction.

The proposed approach resorts to k-step Advantage Actor Critic (A2C) [20] to allocate resources under different IAB network conditions. This technique is applied to our scenario because it can take advantage of both value-based and policy-gradient approaches and it empirically performs better than other similar approaches such as Asynchronous Advantage Actor Critic (A3C) and DQN, as we found out in our preliminary tests. A2C makes sequential action decisions based on the current state of the environment, however, it is difficult to select the appropriate action for the next slot when drastic changes can occur in networks (e.g., dynamic link blockages). To address this issue, LSTM network is adopted to characterize link status variation regularities and feed A2C with a processed state description indicating at which point of the repeating history the environment status currently is.

The essential elements of the RL approach (state space, action space and reward function), the NN's architecture, and their training procedures are further elaborated in the following paragraphs.

5.3.1. State space, action space, and reward function

State Space The factors having an impact on the throughput in an IAB network mainly consist in two elements: the buffer occupation in each IAB-node (i.e., whether relay nodes store enough bits to be transferred) and links status (i.e., whether the links are unobstructed or not). As explained in Section 3, backhaul links are hardly exposed to blockages in practice, therefore we consider in the state definition only the status of access links, which, instead, can undergo several blockages. Thus, our state vector is built from the concatenation of two components: the vector of the number of data bits buffered in each IAB-node (excluding IAB-donor) and the binary vector representing the blockage status of every access link.

The buffer-occupation vector at the end of slot t is an $(|\mathcal{R}_{sub}|)$-dimensional vector storing the number of bits buffered at each IAB-node n, B_n^t, normalized to the number of bits that can be transferred in a slot over the link with the minimum capacity of the whole network, $c_{min} \cdot \delta$. This normalization allows to shrink the state space so as to facilitate the RL agent's exploration, thus accelerating the convergence of the learning process. Namely, the buffer-occupation vector at step t is defined as:

$$S_t^1 = [s_n^t]_{n \in \mathcal{R}_{sub}}, \text{with } s_n^t = \left\lfloor \frac{B_n^t}{c_{min} \cdot \delta} \right\rfloor, \tag{11}$$

while the link-blockage vector can be written as

$$S_t^2 = [o_l^t]_{l \in \mathcal{L}^{ACC}}, \tag{12}$$

where the value of o_l^t depends on the availability of link l in the current slot t. Two assumptions about the knowledge on the link status can be made. In a first more ideal scenario, the status of all access links can be monitored slot-by-slot, which we call *fully-observable* case, and o_l^t is defined as:

$$o_l^t = \begin{cases} 1, & l \text{ is unblocked,} \\ 0, & l \text{ is blocked,} \end{cases} \quad \forall l \in \mathcal{L}^{ACC}. \tag{13}$$

In a second more realistic scenario, we assume that only the status of those links appearing in the pattern selected in the current slot t can be collected, according to whether transmissions are successful or not. We call this scenario *partially-observable* case. Based on this consideration, if an access link is outside the pattern selected, we deem its status as "unknown", therefore we define o_l^t as:

$$o_l^t = \begin{cases} 1, & l \text{ is unblocked, inside pattern,} \\ 0, & l \text{ is blocked, inside pattern,} \\ -1, & l \text{ is outside pattern,} \end{cases} \quad \forall l \in \mathcal{L}^{ACC}. \tag{14}$$

The RL state vector is the concatenation of the two aforementioned vectors: $S_t = [S_t^1, S_t^2]$.

Action Space The link patterns generated in the CG approach of Section 4.2 serve as actions among which the RL agent can select the one to perform in each slot. Each pattern

[4] For a fair comparison with the optimization approach in Section 4, we assume that queue sizes do not limit the performance of the system. However, queue limits can be easily added to the RL environments and the agent can be trained accordingly.

e contains links as sub-actions and the number of bits to be transmitted in a single slot indicated by capacities $B^e_{BH(i,j)}$ and $B^e_{ACC(i,u)}$ for backhaul and access links, respectively. Denoting the set of generated patterns as $\mathcal{E}$, the action at step t is $A_t \in \mathcal{E}$.

In each slot (step) t, the agent selects a pattern A_t according to the current policy π, thus the links within this pattern are activated and enabled to transfer data. Whether or not activated links can truly deliver bits finally depends on whether IAB nodes have enough bits buffered and on the presence of random obstacles obstructing the links.

Reward Function Considering this work aims to maximize the total traffic volume downloaded by UEs in a frame, an intuitive idea is to define the immediate reward as the total number of bits UEs receive in each slot. However, this design strongly biases the solution towards UEs directly connected to the IAB-donor. In such a multi-hop network scenario, data received by UEs in a certain slot are the cumulative result of the bits moved through the wireless backhaul in previous slots, which doesn't produce any immediate transmission to UEs. Also, we need to normalize the reward to improve the agent's learning, this means normalizing the number of transferred bits. In short, we have to provide an answer to the following questions:

- How could we precisely evaluate the current action based only on its immediate effect, while simultaneously considering the cumulative effect of previous actions?
- How could we relate throughput to reward, and also maintain a normalized reward?
- How could we avoid the bias on direct connections between IAB-donor and UEs, which provide a myopic immediate advantage?

Based on these considerations, the reward function at step t is defined as:

$$R_t = \sum_{\substack{(n,u)\in A_t:\\ n\in\mathcal{R}_{sub},u\in\mathcal{U}}} I^t_{n,u}, \tag{15}$$

where $I^t_{n,u}$ is a binary indicator of whether an access link of IAB-node n (IAB-donor excluded) is effective at t. It assumes value 1 when link (n,u) indeed delivers bits (i.e., there are bits in the transmitter's queue and the link is not blocked), 0 otherwise. Therefore, the immediate reward R_t counts the number of access links between IAB-nodes and UEs that effectively transfer data.

5.3.2. *Neural Network Architecture*

The neural network architecture we have designed for the resource allocation problem in IAB networks is sketched in Figure 4. It mainly consists of three key components: a pre-processing network, an actor network and a critic network.

The *pre-processing network* transforms the state vector representing the status of the current slot into a prediction of the IAB network status in the next slot. Specifically, it contains a fully connected layer which extracts features from the input state vector. Then, the LSTM layer captures the regularities in dynamics of IAB nodes' queue lengths and blockage behaviors by constantly updating its hidden state and cell state shown as green and orange circulating flows in Figure 4. LSTM's hidden state and cell state are updated based on forget gate, input gate and output gate, and differently operate[5]. Specifically, the hidden state focuses more on recent experience, while the cell state stores relatively long-term memory with the help of the forget gate to eliminate unessential memories and the input gate to filter in useful fresh information from hidden state and new input. LSTM layer outputs the hidden state to the A2C network, providing a concise prediction about the near-future state, which captures the upcoming situation, even in case of sudden changes, so as to assist A2C in selecting the appropriate action.

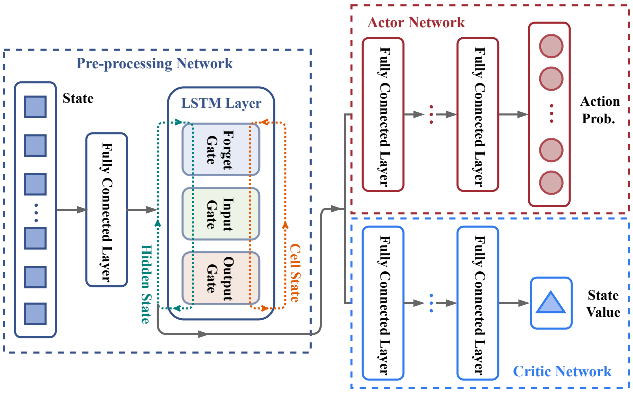


Figure 4: Neural network architecture for IAB scheduling.

Actor network and *critic network* are both composed of several fully connected layers to represent the policy and the value function. The actor network adopts a Softmax output for the action probability distribution, $\pi(A_t|S_t)$, from which the next action to be executed is sampled. The critic network utilizes a linear output for the estimated state-value function, $v(S_t)$. As we can see in Figure 4, the actor network's output dimension is the cardinality of the action space, while the critic network has only one unit for the scalar state value.

5.3.3. *Training Algorithm*

The NN model presented is trained with momentum-based methods and back-propagation through time, relying on the data collected from the interaction process involving the state space, the action space, and the reward function defined in Section 5.3.1. The NN parameters consist in ψ for the critic network to estimate state values, θ for the actor network to generate action probability and ω for the pre-processing network.

We can identify a *critic part* of the framework, which consists of the pre-processing and the critic NN, that aims at minimizing the *value loss*, namely the mean squared error between the current and the estimated state value, formally

[5] We omit the details of the three gates due to limited space. Please, refer to [10] for more details.

expressed as:

$$\min_{\psi,\omega} L_c = (G^k(S_t) - v(S_t;\psi,\omega))^2, \tag{16}$$

$$G^k(S_t) = \sum_{i=0}^{k-1} \gamma^i R_{t+i} + \gamma^k v(S_{t+k}), \tag{17}$$

where the expected k-step return $G^k(S_t)$ is computed at step $t+k$ based on k-step experience.

Considering the policy's performance measure is the long-term reward (critic), the goal of the *actor part* of the framework (pre-processing and actor NN) is to guide the policy parameters θ and ω in the direction of $\nabla_{\theta,\omega}\mathbb{E}[G_t]$ to derive the best action policy $\pi(S_t;\theta,\omega)$. As explained in Section 5.1, a low-variance unbiased estimate of the policy gradient can be considered: $\nabla_{\theta,\omega}\log\pi(A_t|S_t;\theta,\omega)(G^k(S_t) - v(S_t))$. Moreover, in order to promote the action exploration, thus preventing a premature convergence to sub-optimal deterministic policies, the policy entropy $H(\pi(S_t;\theta,\omega))$ is included in the policy error minimization, which is computed as: $-\sum_{A_t}\pi(A_t|S_t;\theta,\omega)\log\pi(A_t|S_t;\theta,\omega)$. In this way, similarly to the value loss, the *policy loss* to be minimized is defined as:

$$\begin{aligned}\min_{\theta,\omega} L_a = &- \log\pi(A_t|S_t;\theta,\omega)(G^k(S_t) - v(S_t)) \\ &- \eta H(\pi(S_t;\theta,\omega)).\end{aligned} \tag{18}$$

Let κ denote the concatenation of ψ, θ and ω, i.e., $\kappa = [\psi,\theta,\omega]$. The ultimate goal of the training process is to iteratively update κ to minimize the total loss function L (19), which is the sum of the policy loss and value loss:

$$\begin{aligned}L =& - \log\pi(A_t|S_t;\kappa)(G^k(S_t) - v(S_t)) - \eta H(\pi(S_t;\kappa)) \\ &+ \beta(G^k(S_t) - v(S_t;\kappa))^2, \end{aligned}\tag{19}$$

$$\kappa = \kappa + \nabla_\kappa L. \tag{20}$$

The parameter vector κ updates via Equation (20).

The detailed learning procedure is described in Algorithm 1, which includes flow routing and pattern scheduling aspects in order to compute the state transition and the reward. The whole training period spans T_{max} steps (slots), while the NN model is updated every t_{max} steps (slots). Considering the IAB network needs to undergo an initial transient period to reach a realistic steady state (i.e., stationary buffer levels and link behaviors), we introduce a warm-up period $t_{warm-up}$.

After the parameter initialization of pre-processing, actor and critic networks (Line 1), the system warms up (Line 3). During the warm-up, rewards are set to 0 to prevent a policy bias in favor of IAB-donor's direct access transmissions. Before each model update, the pointer to the beginning of the train sequence t_{start} is updated and the gradient $d\kappa$ is set to 0 (Line 5).

Data for the main iteration update of gradients and parameters (Lines 4-25) are collected in a data collection loop in Lines 6-17. The data collection loop focuses on an experience of t_{max} steps (stopping if the interaction reaches a terminal state). In each run of the data collection loop, the action A_t is selected according to the current policy $\pi(A_t|S_t;\theta,\omega)$, which is determined by the NN vectors θ and ω (Line 7). Then, the set of blocked links L_b is removed from A_t (Line 9) to simulate the occurrence of blockages according to the model described in Equation (1). Finally, the data transfer is performed over unblocked links. (Lines 10-14). The number of transferred data bits from each transmitter's buffer to each receiver's buffer (Lines 12-13) is limited by:

1. the overall number of bits in transmitter i's buffer at step t: B_i^t (Line 11)
2. the number of bits equally assigned by the node i's transmitter to each of the unblocked parallel links, $(i,\cdot)$, in A_t: $\frac{B_i^t}{|\{(i,\cdot)\}|}$
3. the capacity $c_{i,j}$ of link (i,j) that limits to $c_{i,j}\cdot\delta$ the bits transferred over link (i,j) within a slot.

Effective link data transfers determine the reward, which is set to the number of IAB-nodes' access links that can transfer a non-null number of bits (Line 14). The next state S_{t+1} is updated considering buffer-occupation vector S_{t+1}^1 and link-status vector S_{t+1}^2, which are filled according to Equation (11) and Equation (12) (Lines 15-16). As for the link status, either the fully-observable case or the partial-observable case can be selected.

After the data collection loop, the value of the expected k-step return $G^k(S_t)$ is computed. Its initial value is set in Lines 18-19, then the gradients are computed in Lines 21-24 based on Equations (19) and (20). Finally, NN parameters' vector κ is updated in Line 25. The above operations are repeated until the learning phase ends, after T_{max} steps.

This is an offline training procedure, as in most of the DRL applications, in which a deep NN, trained in a virtual but realistic environment, can be then used in real network operations in a real environment. This training procedure requires some computational effort, however it only needs to be run once to set the appropriate vector κ, then the trained NN can be used, with much less effort, to properly drive the pattern selection in real-time, usually implemented with dedicated neural engine hardware.

5.4. Online Learning

Since the offline training procedure described in the previous section instructs NNs using a random environment, the RL agent can easily deal with the intermittent link availability according to its implicit statistic. However, if this statistic is not fully stationary and some drastic change occurs in the distribution of link blockage probabilities, such pre-trained NNs can incur in a performance degradation.

To tackle drastic changes, a more attractive solution is to perform online learning, which means that NN training and testing are carried out in parallel. In particular, the most-recently updated NN model interacts with the current environment, while in the meantime, the results of these interactions are collected as training data to be used in the next NN model update. In this way, NNs can catch system dynamics on-the-fly and adjust NN weights to adapt to the new environment.

Algorithm 1 *Learning Procedures on IAB Resource Allocation*

Parameters: total training steps T_{max}, model update interval steps t_{max}, warm-up steps $t_{warm-up}$.

```
Initialize neural network weight vector $\kappa = [\psi, \theta, \omega]$;
Initialize step $t \leftarrow 1$;
Warm up the IAB system within $t_{warm-up}$;
while $t < T_{max}$ do
    $t_{start} \leftarrow t$; Get state $S_t$; Reset gradient $d\kappa \leftarrow 0$;
    while $t - t_{start} < t_{max}$ and $S_t$ is not terminal do
        Select pattern $A_t$ based on $\pi(A_t|S_t; \theta, \omega)$;
        $R_t \leftarrow 0$;
        Eliminate blocked link set $A_t \leftarrow A_t \setminus L_b$;
        for $(i, j) \in A_t$ do
            if $B_i^t > 0$ then
                $B_i^t \leftarrow B_i^t - \min\{\frac{B_i^t}{|\{(i,\cdot)\}|}, c_{i,j} \cdot \delta\}$;
                $B_j^t \leftarrow B_j^t + \min\{\frac{B_i^t}{|\{(i,\cdot)\}|}, c_{i,j} \cdot \delta\}$;
                if $i \in \mathcal{R}_{sub}, j \in \mathcal{U}$ then $R_t \leftarrow R_t + 1$;
        Get $S_{t+1}^1$, $S_{t+1}^2$ based on Eqs. (11) and (12);
        $S_{t+1} \leftarrow [S_{t+1}^1, S_{t+1}^2]$;
        $t \leftarrow t + 1$;
    if $S_t$ is not terminal then $G \leftarrow v(S_t; \psi, \omega)$;
    else $G \leftarrow 0$;
    $i \leftarrow t - 1$;
    while $i \geqslant t_{start}$ do
        $G \leftarrow R_i + \gamma G$;
        Update $d\kappa \leftarrow d\kappa + \nabla_\kappa L$ based on Eq. (19);
        $i \leftarrow i - 1$;
    Update $\kappa$ using $d\kappa$ based on Eq. (20);
```

Despite being a promising solution, the online training of LSTM-based NNs presents two big challenges: 1) how to deal with the memory incorporated in the LSTM layer when the environment changes; 2) how such an online learning approach can be realistically implemented in real-time.

5.4.1. Dealing with LSTM memory

The LSTM layer of our framework is in charge of detecting potential regularities in the past link blockage behavior. This is of fundamental importance in stationary conditions to select appropriate actions, however this memory can bias NN training after a sudden environmental change. On the opposite, the complete removal of the LSTM layer would negatively affect the performance as well, due to the lack of good predictions on the network status RL agent's actions will have to face. Therefore, we have investigated on three strategies in order to strike the balance between these two opposite aspects and better adapt the online training to blockage behaviors. Whenever a radical change happens, which can be identified by a sharp drop in the cumulative reward, we apply the following strategies:

- *Reset-all*: Reset the memory factors in the LSTM layer (i.e., the cell state and hidden state) and all the parameters in the whole NN.
- *Reset-memory*: Reset only the memory factors in the LSTM layer (i.e., the cell state and hidden state) and leave all the other parameters the same.
- *Reset-none*: Make no changes to the entire NN model.

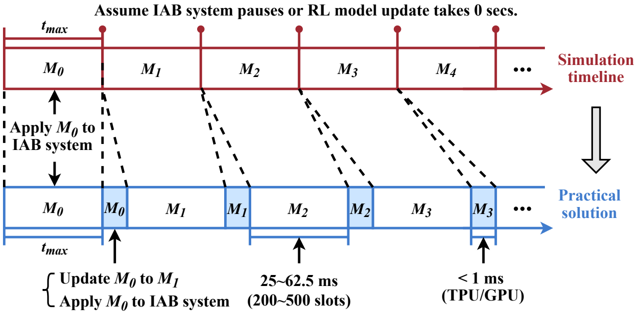


Figure 5: Ideal simulation timeline and feasible solution in practice.

In the simulation experiments, we compare the results obtained with these three strategies to understand which one performs best.

5.4.2. Dealing with implementation and feasibility

In the offline training of Algorithm 1, we select the best actions according to the current NN model and apply them to the IAB network in the virtual environment for t_{max} steps (slots) to collect a new training data batch. Then, we use this data batch as new input for the NN whose parameters has to be updated. The updated NN model will be used in the next round of t_{max} steps to collect further data and the process repeats until the end of the training is reached.

This approach is applicable only to an offline learning scheme, with no training interactions with the real IAB network. Indeed, a mere application of Algorithm 1 to an online learning paradigm would require either the update time of NN parameters to be negligible or the IAB network to pause and wait for the NN model to be updated, which are two approaches not always possible in practice. To address this issue of the online training, we propose the following scheme, sketched in Figure 5. We refer to an ideal simulation timeline (drawn in red) in which the data collected during a batch of t_{max} interactions by using NN model M_x are processed at the end of the batch to immediately update the NN model and obtain the new model M_{x+1}, which will be used in the next batch. In the practical solution (drawn in blue), we introduce transient periods (shaded) during which the new NN parameters are computed and, in the meantime, the old model M_x is applied, but data are not collected. When a transient period ends, the updated NN model M_{x+1} is put to use and a new batch of t_{max} interactions is collected. In doing so, the IAB network can keep running while the model can be trained as well.

The impact of these transient periods on the convergence speed is limited. Indeed, the NN model update procedure requires less than 100 ms on our general-purpose laptop. Therefore, with the help of optimized coding and specialized hardware, like TPUs and GPUs [32, 29], we can reasonably assume to have an improvement of a factor 100, which makes the update procedure last less than 1 ms, namely less than 8 slots of a physical IAB frame. This is a very small time period if compared with the data collection time t_{max} (in our experimental settings, 200 ∼ 500 steps/slots, corresponding to 25 ∼ 62.5 ms). In this sense, the online-learning training period will not be significantly extended.

6. Experimental Results

In this section, we evaluate the performance of our LSTM-assisted A2C-based resource allocation method through a simulation campaign. We first describe the considered network scenario and the NN model settings. Then, the performance of offline and online schemes is analyzed.

6.1. IAB Network Scenario

The instances we consider are the results of an internally-developed random instance generator compliant with 3GPP NR IAB simulation guidelines [2]. The playground consists of a $300m \times 300m$ square with 1 IAB-donor, 4 IAB-nodes and 30 UEs randomly deployed. The IAB-donor is placed at a height of 25m and is equipped with a single 24x16 panel antenna array that can generate and process 4 simultaneous streams. IAB-nodes are placed at a height of 6m and are equipped with 4-panel 8x6 antenna array with elements per panel, each able to create and process 1 stream. UEs are equipped with omni-directional antennas and set to a height of 1.5m. The maximum transmission power is set to 32dBm for the IAB-donor and 23dBm for each IAB-node panel.

We consider a 3GPP NR TR 38.901 channel model [1]. IAB access and backhaul transmissions are carried out at 28 GHz with 400 MHz of bandwidth and NR Numerology #3 (120 kHz subcarrier spacing). Each frame consists of 80 slots, each of which lasts 125 μs. We consider a single MCS for backhaul and access: 16 and 8, respectively[6]. MCS 16 corresponds to a SINR threshold of 5.60 dB and rate of 525.9 Mbps, while MCS 8 can achieve a rate of 121.4 Mbps with a SINR threshold of -3.77 dB.

6.2. NN Settings

In the experiments, the same NN settings are used in both offline and online approaches. In particular, the NN model is composed of 1 fully connected layer with 32 units, 1 LSTM layer with 64 hidden units, 8 fully connected layers with 32 units for both actor and critic networks. The output layers of actor and critic networks use Softmax and linear functions.

Reward's discount factor γ is set to 0.99, hence a long-term reward is considered. Weights η for policy entropy and β for value loss in total loss function of Equation (19) are set to 0.01 and 0.25. Learning rate and batch size for the learning process are 0.007 and 200, respectively. RMSProp Optimizer is used to minimize the total loss so as to adjust the NN parameters.

The total time for the offline training spans 1.8e6 steps, which takes approximately 40 minutes on our Intel(R) Xeon(R) CPU E5-2640 v4 @ 2.40GHz and 125GB RAM machine. Since it takes some time to reach a stationary situation where IAB-nodes' queues have enough data to be transferred, we have considered a warm-up period of 400 slots, empirically determined, during which rewards are set to 0. This allows us to avoid biasing the solution on IAB-donor access transmissions, as they can provide immediate initial reward with respect to IAB-nodes' ones which initially have empty queues.

The online approach is evaluated in a dynamic IAB network scenario where not only a different number of links undergo random blockages, but also blockage dynamics is totally changed several times. In particular, the results of the offline approach are compared with those of the online approach, demonstrating that the online training solution can mostly outperform the offline model via automatically adjusting NN weights in accordance with blockage dynamics.

6.3. Offline Approach Performance Analysis

In this part, the LSTM-A2C NN models (referred to as DRL in the following) are trained offline in scenarios characterized by the same link blockage statistics as in the testing scenarios to which the trained NN model is subsequently applied. DRL's performance is compared against three alternative resource allocation methods:

- CG-RND: where the link pattern to be activated in the current slot is randomly chosen among those available at the end of pricing process in CG Section 4;
- CG-OPT: where the (quasi-)optimal frame is computed by using the whole CG approach, which considers ideal fully-reliable links.
- Multi-Slot: a heuristic algorithm proposed in [26] to perform link scheduling, which coordinates the link interference to construct sets of links (similar to the idea of *link patterns* in our work) that satisfy SINR conditions. According to the Multi-Slot algorithm, we generate patterns and then iteratively apply them in sequential order slot by slot.

All the values are based on the average of 10 instances randomly generated.

All four methods are radically different and have different complexity levels, which are difficult to precisely compare. Therefore, we resort to their average solution time on our laptop to have an approximate idea. CG-RND solves a sequence of integer and linear programming models and stops when no objective function improvement can be obtained. Each instance requires a different number of iterations, but we have experimentally checked that the improvement is limited beyond the 200-th iteration. The entire procedure lasts less than 1 minute. The optimization approach of CG-OPT needs more time for the final integer solution compared with CG-RND, thus its total time increases up to more than 10 minutes. Dealing with resource allocation problems over integer resources, both CG-RND

[6] Note that we have considered a single MCS for access and backhaul only for sake of simplicity. Indeed, MCS selection is not in the charge of the RL agent, whose actions are the different link patterns. MCS is implicitly included in the definition of link patterns, which are automatically generated in the CG procedure.

Table 3
Random blockage distribution settings for offline models.

Block. types	Light	Mid	Severe
μ	5.58	7.37	7.88
$1/\lambda$	5000	3500	2300

Table 4
Average per-UE data rates (Mbps).

Block. types	Light	Mid	Severe
CG-RND	25.756	22.052	16.110
CG-OPT	38.588	26.843	17.598
DRL	36.821	28.802	21.752

and CG-OPT are NP-hard. The Multi-Slot algorithm is a greedy polynomial algorithm with a worst-case complexity of $O(n^3)$, where n is the number of links. It is very fast and takes about 50 ms to provide a solution. All these methods provide the final frame structure at the end of their execution. Our DRL approach is based on a NN formed by totally about 1000 units in 20 layers. It needs about 40 minutes to be trained, which is done once for all, and it takes about 1 ms to be applied to define each slot of the frame. Clearly, simple approaches have a time advantage, but their performance has strong limitations, which we discuss later on.

Three levels of link-blockage intensities are imposed on IAB networks. They are defined by three sets of parameters μ, σ, λ of the distributions in Equation (1), whose values are shown in Table 3. We set σ to 0.5 ms and adjust μ to obtain ratios of average blocked duration to average non-blocked duration of respectively 0.06, 0.51, and 1.30. They correspond to increasing blockage intensities, hence we refer to them as *light*, *mid* and *severe* blockages in the rest of the article. Note that, since we have noticed a fast convergence of the DRL approach, we have accelerated the blockage model of a factor 10, for both online and offline versions, to shorten the simulation duration.

A first result is related to the two different state spaces described in Section 5.3.1, which consider either the status of all access links (fully-observable) or only that of those links activated in the currently selected pattern (partially-observable). We performed all the experiments with both alternatives and the results were very similar. Therefore, for the sake of brevity, we show in this article only the ones using the partially-observable state, which refers to a more realistic approach.

6.3.1. IAB Network Traffic Delivery

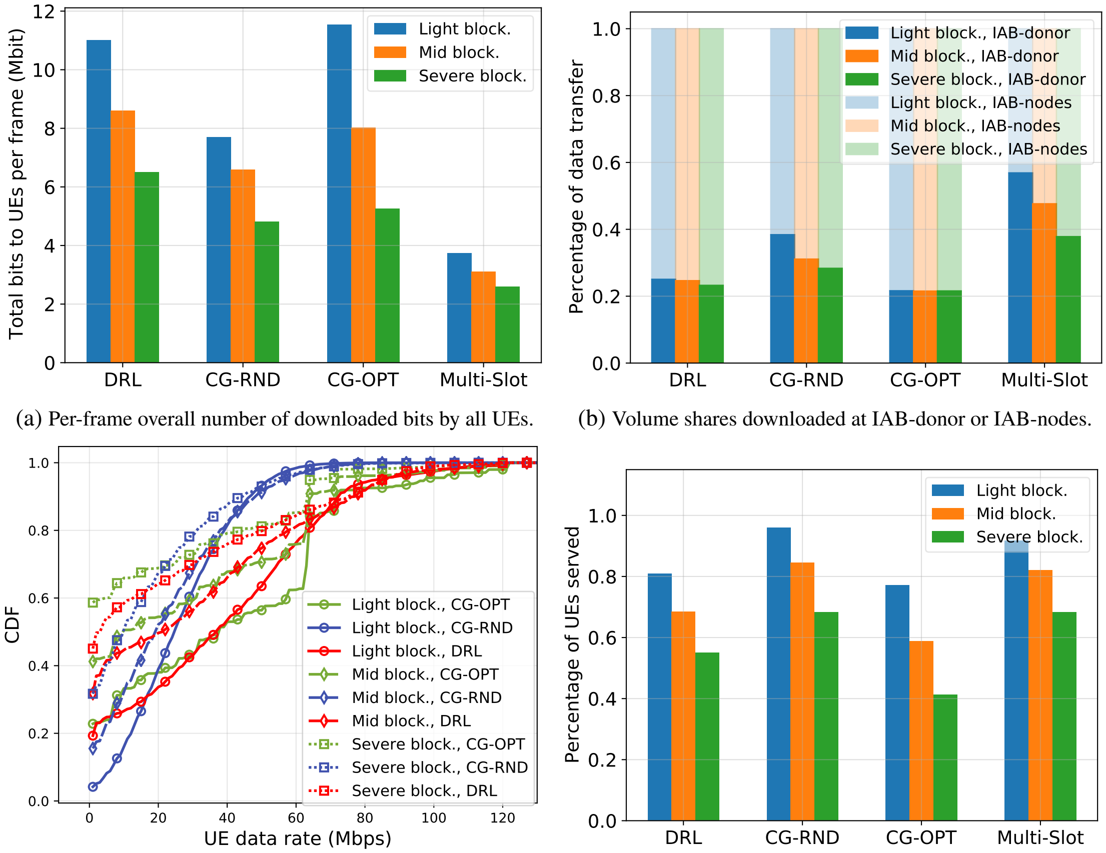
The performance of the four methods can be evaluated considering the average overall traffic volume (number of bits) delivered to all UEs in a frame, which corresponds to the value of the objective function (2a). In Figure 6(a), the differences among the four methods over the three blockage levels are illustrated. In general, the total traffic volume decreases as more intense blockages are introduced into the scenario. Multi-Slot has a remarkably low performance because (1) the quality of its generated patterns is lower than that of CG-RND, even if CG-RND schedules its CG-generated patterns randomly at each slot, which implies patterns must be carefully designed; (2) the path routing and scheduling also need to be optimized in addition to patterns, this is the reason why CG-OPT performs much better, in particular under light blockage conditions; (3) adaptive schemes are essential for a scheduling method to be applied in dynamic networks, as shown by the comparison with DRL; (4) Multi-Slot does not consider queue lengths of IAB-nodes that is important to define a good pattern sequence in a frame. Comparing CG-OPT, CG-RND and DRL, the order of the throughput reduction is CG-OPT > DRL > CG-RND, which is due to the fact that the CG-OPT method, working under ideal link behavior assumptions, is more severely affected by blockages. By contrast, the CG-RND method, using all patterns with the same probability, is less influenced by blockages as the inefficiencies in the multi-hop delivery tend to dominate over blockages, i.e., the limited number of delivered bits is due more to the recurrent lack of bits in the transmission buffer when a link is activated than to a blockage that occasionally prevents the link from transmitting.

Figure 6(a) also shows a fundamental aspect of our approach: DRL delivers more bits than CG-OPT after the blockage intensity increases over a certain level (e.g., settings for mid and severe blockages). Thanks to the adaptability of RL to a dynamic environment, the DRL method, assisted by LSTM, can learn the regularities in access links' availability during the training phase and select the appropriate link-pattern sequence accordingly. The CG-OPT method, which provides the best choices in ideal conditions, can still perform well when light blockages make the scenario quasi-ideal. However, as soon as the blockage intensity impairs this ideality, its performance quickly decreases. The effect on perceived UE throughput can also be evaluated by the average data rates reported in Table 4.

Figure 6(b) provides an insight into how the data bits are delivered to UEs via the IAB network. The upper translucent bars represent the traffic volume percentage received by UEs from IAB-nodes through multi-hops, while the lower solid bars report the complementary volume percentage directly received from the IAB-donor. We can see that DRL and CG-OPT methods can more efficiently utilize the hops of the wireless backhaul. This efficiency is almost independent of the blockage intensity, demonstrating that a smart resource allocation is necessary to operate a multi-hop wireless backhaul in any conditions. Indeed, the CG-RND and Multi-Slot methods result in more bits directly sent by the IAB-donor, which reduce when the blockage intensity increases only because UEs can be reached by more than one IAB-node but only one IAB-donor, and thus, IAB-node delivery is more robust than IAB-donor's one. Moreover, CG-RND delivers less bits directly from IAB-donor than Multi-Slot because good patterns allow to better exploit the wireless backhaul.

6.3.2. UE's Quality of Service

The cumulative distribution function (CDF) curves of achievable UEs' data rates are shown in Figure 6(c). Since previous analysis has shown much worse performance of



(a) Per-frame overall number of downloaded bits by all UEs.

(b) Volume shares downloaded at IAB-donor or IAB-nodes.

(c) Per-UE data rate CDF.

(d) Percentage of served UE.

Figure 6: Performance of offline DRL approach considering different blockage intensities, compared with CG-RND, CG-OPT and Multi-Slot.

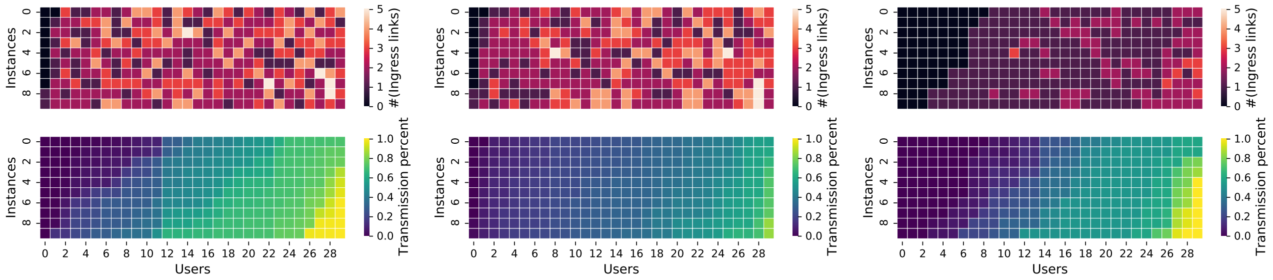
Multi-Slot than that of DRL, we don't further include Multi-Slot in CDF curves to simplify the presentation of the results. As can be seen from the upper right corner of the figure, under three cases of different blockage densities (i.e., light, mid and severe blockages), the maximum rates achieved by DRL and CG-OPT are near 121.4 Mbps (the maximum achievable rate of MCS 8), while CG-RND can at most reach about 60 Mbps. This means that the problem faced is not trivial and only a careful link-pattern selection can allow us to reach good performance. In addition, DRL can learn how to provide high throughput, even in case of blockages, to those users that are not directly affected by them. The average per-UE data rates are reported in Table 4: while in light-blockage conditions the throughput provided by DRL is only 4.5% less than CG-OPT, DRL outperforms CG-OPT by more than 23% in case of severe blockages.

The values of the CDFs at the origin indicate the percentage of users that cannot be served. This information is better described by Figure 6(d), which indicates the percentage of UEs in the playground that receives a non-null throughput in the different scenarios. The first and expected result is that as the blockages get denser, more users are excluded from the service. Two interesting aspects further emerge: 1) CG-RND and Multi-Slot methods show the highest service percentages, because they do not tend to select the best UEs to maximize the overall throughput, but rather to reach all UEs with the same probability, although with a small throughput. 2) DRL can serve more users than CG-OPT in all three blockage situations, showing an advantage not only from a throughput perspective but also in terms of coverage, when the links' availability is not close to the ideal.

6.3.3. DRL success analysis

We analyze now the reasons of the success of DRL method with the help of Figure 7, where, for the sake of brevity, we consider only the case of severe blockages, but the same considerations apply to other cases as well. The heatmaps in the top row show the number of different access links incident to each UE (horizontal axis) scheduled in the whole testing period of each instance (vertical axis). The bottom row instead shows the percentage of slots in which the indicated UE appears as a receiver of a link in the activated patterns. For each method, the order of instances' IDs and UEs' IDs in every heatmap is computed according to the descending order of slot percentage.

Comparing the three heatmaps in the top row, we can see that CG-OPT method activates the smallest number of different access links for each UE, as shown by its darkest heatmap. DRL and CG-RND, instead, resort to more access links, which increase the probability to use alternatives when an access link is blocked. This increases the scheduling robustness. In the bottom row, the lightest heatmap shows that

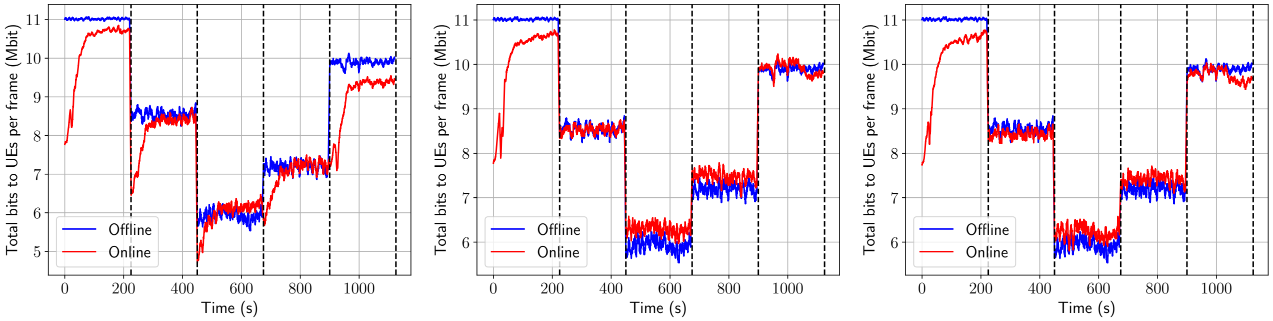


(a) DRL.

(b) CG-RND.

(c) CG-OPT.

Figure 7: Comparison of the three methods with severe blockages.



(a) Total Traffic Volume - Reset-all.

(b) Total Traffic Volume - Reset-memory.

(c) Total Traffic Volume - Reset-none.

Figure 8: Comparison among the three online schemes in Lo-Hi-Lo-blockage scenario.

the DRL's access link diversity is obtained by selecting patterns with more access links than CG-OPT and CG-RND, this stresses again the idea that the best scheduling strategy is to select patterns with redundant access links so that the probability that at least one is effective is higher.

6.4. Online Model Performance Analysis

We conclude this section by testing the online training framework, where a NN model is continuously trained while being applied to an IAB network. Since the ability of our approach to adapt to random blockages and provide high throughput has already been shown in the previous paragraphs, we intend to evaluate here whether the online model keeps learning from the ongoing interactions with the IAB network and can automatically re-adapt on-the-fly to changing dynamics. All the results presented in this section are the averages over 10 random instances and we apply a moving average of 0.0625s to all reported plots.

In the following tests, the blockage intensities are divided into 5 levels from lightest to severest (from 1st to 5th level, correspondingly) which is defined by the parameters in Table 5 referring to the blockage model in Equation (1). As in the offline case, σ is still set to 0.5 ms for simplicity. The tests are conducted in two representative scenarios:

- *Lo-Hi-Lo-blockage*: where the experiment begins with the lightest (1st-level) blockage intensity and proceeds with 3rd, 5th, 4th, and 2nd-level blockage intensities;
- *Hi-Lo-Hi-blockage*: where the experiment begins with the severest (5th-level) blockage intensity and proceeds with 3rd, 1st, 2nd, and 4th-level blockage intensities.

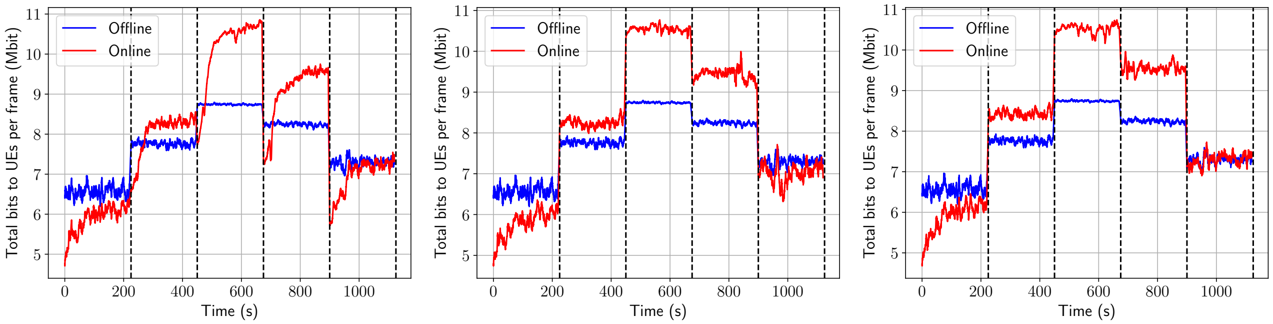
Table 5
Distribution settings for blockage levels in the online framework.

Levels	1st	2nd	3rd	4th	5th
μ	5.58	6.83	7.37	7.66	7.88
$1/\lambda$	5000	4250	3500	2900	2300

The three online strategies (reset-all, reset-memory, and reset-none) presented in Section 5.4 are tested in these two scenarios.

In Figures 8 and 9, the results show the performance in terms of total traffic volume delivered to UEs in a frame. The horizontal axis indicates the timeline under the simplified, but reasonable, as shown in Section 5.4.2, assumption of negligible NN update time. In these figures, the blue curves represent the performance in the testing period of the pre-trained offline model, while the red curves describe the system behaviors during online training. Vertical dashed lines delimit the stages of the experiment by indicating sudden changes of blockage intensities. Moreover, while the offline model is pre-trained over a scenario with the same statistics as those of the first stage, the online model starts from a random initialization and converges to a stable performance by the end of the first stage.

As a first observation, we can see that the more the stage blockage intensity differs from the one of the first stage, the higher performance advantage the online model takes over the offline one. This confirms that faced scenarios are indeed different and, ideally, different NN weights would be required. Another interesting finding is that the online model has only a small lead over the offline model in the Lo-Hi-Lo-blockage scenario, while in the Hi-Lo-Hi-blockage scenario,



(a) Total Traffic Volume - Reset-all.

(b) Total Traffic Volume - Reset-memory.

(c) Total Traffic Volume - Reset-none.

Figure 9: Comparison among the three online schemes in Hi-Lo-Hi-blockage scenario.

the online model largely outperforms the offline model. The reason is that frequent blockages of the first stage of the Hi-Lo-Hi-blockage scenario implicitly set a limit to the offline model's policy, which leaves more rooms for the online model to improve over when more link opportunities become available in the next stages.

This allows us to draw further conclusions. If we want to use an offline-training approach, we have to pay attention to training it in an environment that is less affected by blockages than it can potentially be when severer conditions are met. However, the blockage intensity cannot be too light, otherwise blockage countermeasures cannot be learned. Vice versa, an online-training solution can always perform best.

Finally, the comparison across three strategies (i.e., reset-all, reset-memory and reset-none) indicates that the online approach can potentially perform well. Considering the longer convergence time at each stage of the "reset-all" strategy and the performance similar to the other two strategies, we suggest to use the "reset-none" strategy as the best tradeoff between performance and complexity.

7. Conclusion

In this paper, we have proposed a CG-based DRL approach for resource allocation in mmWave 5G IAB networks able to face realistic link blockages. The results have shown that we can outperform the optimization approaches typically used in wireless multi-hop networks, demonstrating that our approach can automatically adapt to environmental changes. In addition, we have developed an online version of our approach to increase its robustness in front of dramatically changing environments. Indeed, it can catch system dynamics on-the-fly and adjust the training to adapt to the change. We have also carried out an analysis of implementation and feasibility issues of our approach, which has proven how it can be practically implemented just relying on realistic hardware requirements.

Finally, we believe that our approach can be seen as one of the examples in which traditional optimization techniques and recent RL approaches can positively coexist and provide remarkable advantages by synergically leveraging their respective strengths.

Acknowledgment

This work was partially supported by China Scholarship Council (CSC) Grant No. 201806470077.

References

[1] 3GPP, a. Study on channel model for frequencies from 0.5 to 100 GHz, TR 38.901.

[2] 3GPP, b. Study on integrated access and backhaul, TR 38.874.

[3] Capone, A., Carello, G., Filippini, I., Gualandi, S., Malucelli, F., 2010a. Routing, scheduling and channel assignment in wireless mesh networks: optimization models and algorithms. Ad Hoc Networks 8, 545–563.

[4] Capone, A., Carello, G., Filippini, I., Gualandi, S., Malucelli, F., 2010b. Solving a resource allocation problem in wireless mesh networks: A comparison between a CP-based and a classical column generation. Networks: An International Journal 55, 221–233.

[5] Capone, A., Filippini, I., Gualandi, S., Yuan, D., 2013. Resource optimization in multi-radio multi-channel wireless mesh networks, Wiley Online Library.

[6] Du, J., Onaran, E., Chizhik, D., Venkatesan, S., Valenzuela, R.A., 2017. Gbps user rates using mmwave relayed backhaul with high-gain antennas. IEEE Journal on Selected Areas in Communications 35, 1363–1372.

[7] Feng, M., Mao, S., 2019. Dealing with limited backhaul capacity in millimeter-wave systems: a deep reinforcement learning approach. IEEE Communications Magazine 57, 50–55.

[8] García-Rois, J., Banirazi, R., González-Castaño, F.J., Lorenzo, B., Burguillo, J.C., 2018. Delay-aware optimization framework for proportional flow delay differentiation in millimeter-wave backhaul cellular networks. IEEE Transactions on Communications 66, 2037–2051.

[9] He, Z., Mao, S., Kompella, S., Swami, A., 2015. Minimum time length scheduling under blockage and interference in multi-hop mmwave networks, in: 2015 IEEE Global Communications Conference (GLOBECOM), IEEE. pp. 1–7.

[10] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Computation 9, 1735–1780.

[11] Hu, Q., Blough, D.M., 2017. Relay selection and scheduling for millimeter wave backhaul in urban environments, in: 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), IEEE. pp. 206–214.

[12] Kilpi, J., Seppänen, K., Suihko, T., Paananen, J., Chen, D.T., Wainio, P., 2017. Link scheduling for mmwave WMN backhaul, in: 2017 IEEE International Conference on Communications (ICC), IEEE. pp. 1–7.

[13] Lei, W., Ye, Y., Xiao, M., 2020. Deep reinforcement learning based spectrum allocation in integrated access and backhaul networks. IEEE Transactions on Cognitive Communications and Networking .

[14] Li, R., Wang, C., Zhao, Z., Guo, R., Zhang, H., 2020. The lstm-based advantage actor-critic learning for resource management in network slicing with user mobility. IEEE Communications Letters .

[15] Li, Y., Luo, J., Xu, W., Vucic, N., Pateromichelakis, E., Caire, G., 2017a. A joint scheduling and resource allocation scheme for millimeter wave heterogeneous networks, in: 2017 IEEE Wireless Communications and Networking Conference (WCNC), IEEE. pp. 1–6.

[16] Li, Y., Pateromichelakis, E., Vucic, N., Luo, J., Xu, W., Caire, G., 2017b. Radio resource management considerations for 5G millimeter wave backhaul and access networks. IEEE Communications Magazine 55, 86–92.

[17] Luo, J., Rosenberg, C., Girard, A., 2010. Engineering wireless mesh networks: joint scheduling, routing, power control, and rate adaptation. IEEE/ACM Transactions on Networking 18, 1387–1400.

[18] Ma, Z., Li, B., Yan, Z., Yang, M., 2020. Qos-oriented joint optimization of resource allocation and concurrent scheduling in 5G millimeter-wave network. Computer Networks 166, 106979.

[19] MacCartney, G.R., Rappaport, T.S., Rangan, S., 2017. Rapid fading due to human blockage in pedestrian crowds at 5G millimeter-wave frequencies, in: 2017 IEEE Global Communications Conference (GLOBECOM), IEEE. pp. 1–7.

[20] Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K., 2016. Asynchronous methods for deep reinforcement learning, in: International Conference on Machine Learning (ICML), pp. 1928–1937.

[21] Naparstek, O., Cohen, K., 2018. Deep multi-user reinforcement learning for distributed dynamic spectrum access. IEEE Transactions on Wireless Communications 18, 310–323.

[22] Niu, Y., Ding, W., Wu, H., Li, Y., Chen, X., Ai, B., Zhong, Z., 2019. Relay-assisted and QoS aware scheduling to overcome blockage in mmwave backhaul networks. IEEE Transactions on Vehicular Technology 68, 1733–1744.

[23] Niu, Y., Gao, C., Li, Y., Su, L., Jin, D., Zhu, Y., Wu, D.O., 2016. Energy-efficient scheduling for mmwave backhauling of small cells in heterogeneous cellular networks. IEEE Transactions on Vehicular Technology 66, 2674–2687.

[24] Ortiz, A., Asadi, A., Sim, G.H., Steinmetzer, D., Hollick, M., 2019. Scaros: A scalable and robust self-backhauling solution for highly dynamic millimeter-wave networks. IEEE Journal on Selected Areas in Communications .

[25] Polese, M., Giordani, M., Roy, A., Castor, D., Zorzi, M., 2018. Distributed path selection strategies for integrated access and backhaul at mmwaves, in: 2018 IEEE Global Communications Conference (GLOBECOM), IEEE. pp. 1–7.

[26] Saad, M., Abdallah, S., 2019. On millimeter wave 5G backhaul link scheduling. IEEE Access 7, 76448–76457.

[27] Saha, C., Afshang, M., Dhillon, H.S., 2018. Bandwidth partitioning and downlink analysis in millimeter wave integrated access and backhaul for 5G. IEEE Transactions on Wireless Communications 17, 8195–8210.

[28] Sahoo, B., Yao, C.H., Wei, H.Y., 2017. Millimeter-wave multi-hop wireless backhauling for 5G cellular networks, in: 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), IEEE. pp. 1–5.

[29] Shi, S., Wang, Q., Xu, P., Chu, X., 2016. Benchmarking state-of-the-art deep learning software tools, in: 2016 7th International Conference on Cloud Computing and Big Data (CCBD), IEEE. pp. 99–104.

[30] Vu, T.K., Bennis, M., Debbah, M., Latva-Aho, M., Hong, C.S., 2018a. Ultra-reliable communication in 5G mmwave networks: a risk-sensitive approach. IEEE Communications Letters 22, 708–711.

[31] Vu, T.K., Liu, C.F., Bennis, M., Debbah, M., Latva-Aho, M., 2018b. Path selection and rate allocation in self-backhauled mmwave networks, in: 2018 IEEE Wireless Communications and Networking Conference (WCNC), IEEE. pp. 1–6.

[32] Wang, Y.E., Wei, G.Y., Brooks, D., 2019. Benchmarking TPU, GPU, and CPU platforms for deep learning. arXiv preprint arXiv:1907.10701 .

[33] Yang, G., Xiao, M., Al-Zubaidy, H., Huang, Y., Gross, J., 2018. Analysis of millimeter-wave multi-hop networks with full-duplex buffered relays. IEEE/ACM Transactions on Networking (TON) 26, 576–590.

[34] Yuan, D., Lin, H.Y., Widmer, J., Hollick, M., 2018. Optimal joint routing and scheduling in millimeter-wave cellular networks, in: 2018 IEEE Conference on Computer Communications (INFOCOM), IEEE. pp. 1205–1213.

[35] Zhang, B., Devoti, F., Filippini, I., 2020. RL-based resource allocation in mmwave 5G IAB networks, in: 2020 IEEE Mediterranean Communication and Computer Networking Conference (MedComNet), IEEE. pp. 1–8.

[36] Zhuang, H., Chen, J., Wu, D.O., 2017. Joint access and backhaul resource management for ultra-dense networks, in: 2017 IEEE International Conference on Communications (ICC), IEEE. pp. 1–6.

Bibo Zhang received the B.S. degree in information engineering and the M.S. degree in electronics and communication engineering from Beijing University of Posts and Telecommunications, China, in 2015 and 2018. She is currently pursuing the Ph.D. degree in information technology, from Politecnico di Milano, Italy. Her current research interests include resource management in 5G mmWave networks.

Francesco Devoti received the B.S., and M.S. degrees in Telecommunication Engineering, and the Ph.D. degree in Information Technology from the Politecnico di Milano, in 2013, 2016, and 2020 respectively. He is currently a research scientist at NEC Laboratories Europe. His research interests include millimeter-wave technologies in 5G networks.

Ilario Filippini received B.S. and M.S. degrees in Telecommunication Engineering and a Ph.D in Information Engineering from the Politecnico di Milano, in 2003, 2005, and 2009, respectively. He is currently an Associate Professor with the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano. His research interests include planning, optimization, and game theoretical approaches applied to wired and wireless networks, performance evaluation and resource management in wireless access networks, and traffic management in software defined networks. On these topics, he has published over 60 peer-reviewed articles. He serves in the TPC of major conferences in networking and as an Associate Editor of *IEEE Transactions on Mobile Computing* and *Elsevier Computer Networks*. He is an IEEE Senior Member.

Danilo De Donno received the B.Sc. and M.Sc. degrees (cum laude) in telecommunication engineering from Politecnico di Milano, Italy, in 2005 and 2008, respectively, and the Ph.D. degree in Information Engineering from the University of Salento, Lecce, Italy, in 2012. He was a Post-Doctoral Fellow with the ElectroMagnetic Lab Lecce (EML2) of the University of Salento from 2012 to 2015 and a Post-Doc Researcher with the IMDEA Networks Institute, Madrid, Spain, from 2015 to 2017. In July 2017, he joined the Huawei Research Center in Milan, Italy, as a Wireless System Engineer. His areas of interest lie in mm-Wave communications, with main research focus on the development of PHY and MAC algorithms for hybrid and full-digital system architectures.