Delay-Aware Link Scheduling in IAB Networks with Dynamic User Demands

Yekaterina Sadovaya, Olga Vikhrova, Wei Mao, Shu-ping Yeh, Omid Semiari,
Hosein Nikopour, Shilpa Talwar, and Sergey Andreev

***Abstract*—Integrated Access and Backhaul (IAB) is a cost-efficient network densification technology for improving the coverage and capacity of the millimeter-wave (mmWave) cellular networks. In IAB systems, user traffic is forwarded to/from the wired base station by one or more relay stations, known as IAB nodes. Due to the multi-hop relaying, these systems may be subject to large packet delays and poor performance when the load is unevenly distributed among nodes. Addressing this limitation via delay-aware access and backhaul link scheduling in IAB networks is challenging due to potentially large network scale, complex topology, half-duplex, and interference constraints.**

In this paper, the topical link scheduling problem is formulated as a Markov decision problem (MDP) for a single-donor IAB system with a general topology that allows for users with different delay requirements and traffic dynamics. The proposed link scheduling strategy jointly optimizes (i) user traffic routing and (ii) multiplexing of access and backhaul links under half-duplex constraints and non-negligible interference that may arise in dense IAB systems even with high beam directionality. To address the complexity of our formulated MDP, we consider several approximation methods, namely, Q-learning, Monte Carlo Tree Search (MCTS), and genetic algorithms (GAs). Then, we propose a customized version of the GA, which provides the preferred optimality–complexity trade-off and offers a 15% packet delay reduction as compared to the state-of-the-art backpressure algorithm.

***Index Terms*—IAB, millimeter-wave, link scheduling, routing, half-duplex constraint, interference, user dynamics**

I. Introduction

A. Research Motivation

INTEGRATED ACCESS AND BACKHAUL (IAB) technology proposed by the Third Generation Partnership Project (3GPP) in [1] has received substantial attention from both academia and industry as a cost-efficient solution for extending the coverage and improving the performance of future cellular networks operating at mmWave frequencies [2]. Due to offering larger bandwidths than sub-6 GHz systems, mmWave radio is crucial for the fifth generation and beyond (5G/B5G) systems to meet the anticipated traffic growth and more stringent requirements of emerging interactive and immersive applications [3]. However, mmWave links are known to have significantly higher path and penetration losses and are more prone to atmospheric absorption as compared to, e.g., microwave links [4]. Even though the impact of losses can be effectively reduced by using advanced signal processing and multiple-input multiple-output (MIMO) communications to form highly directional links, the resulting coverage is inferior to that of sub-6 GHz deployment. Hence, mmWave systems require ultra-dense base station deployments to alleviate coverage gaps caused by blockage and directional transmissions. The straightforward densification by increasing the number of 5G New Radio (NR) nodeBs (gNBs) per square meter is costly and challenging in some locations. Instead, IAB offers rapid and low-cost on-demand network densification by deploying multiple IAB nodes where additional coverage or capacity is needed.

IAB nodes are wireless relays interconnected with each other and with a donor gNB (DgNB) over wireless backhaul links. Therefore, any new IAB node can be quickly added to the existing deployment, while the already deployed nodes can be moved to a new location. According to the IAB architecture, which is defined in 3GPP TS 38.401 [5], an IAB node accommodates both distributed unit (DU) and mobile termination (MT) functions as shown in Fig. 1. MT function determines IAB node as a child node that is controlled by the other IAB nodes or donor, while DU function makes IAB node behave as a parent for the other IAB nodes and user equipment (UE). IAB nodes and DgNBs can form directed acyclic graph (DAG) and spanning tree topologies according to 3GPP TR 38.874 [1]. Following the principles of the open radio access network (O-RAN), central unit (CU) and DU can utilize various intelligent microservices (rApps or xApps) implemented at the non-real-time and near-real-time radio access network (RAN) intelligent controllers (RICs) [6] to, e.g., predict traffic demands, optimize topology, manage routes, and allocate resources [7], [8] in IAB networks.

While 3GPP specifications consider the possibility of implementing backhauling in out-of-band mode, multiplexing access and backhaul links at the same frequency provides attractive cost benefits due to hardware and frequency reuse. This also makes IAB nodes lower in price than conventional relays due to the reuse of most of the 5G access interfaces at the backhaul links [9]. Moreover, IAB systems can significantly improve network capacity and delay performance due to dynamic time-division duplexing (TDD) configuration, which is enabled by the flexible 5G NR *frame pattern* structure. The performance of IAB networks is reliant on the frame pattern established by the CU and announced by the donor at the beginning of every frame. This pattern indicates the sequence of uplink (UL) and downlink (DL) slots for the donor in a frame and cannot be changed until the end of this frame. Frame pattern along with

Y. Sadovaya (corresponding author), O. Vikhrova, and S. Andreev are with Tampere University, Finland. S. Andreev is also with Brno University of Technology, Brno, Czech Republic. Email: firstname.lastname@tuni.fi

W. Mao, S.-p. Yeh, O. Semiari, H. Nikopour, and S. Talwar are with Intel Corporation, Santa Clara, CA, USA. Email: firstname.lastname@intel.com

user scheduling and routing algorithms provide more detailed instructions for IAB nodes on how time slots can be used to prevent half-duplex mismatch and interference [10].

Despite the promising benefits of the IAB technology, IAB networks are subject to fundamental constraints of wireless multi-hop networks. First, in-band operation introduces several challenges for optimizing resource allocation and link scheduling due to the half-duplex constraint. The latter does not allow IAB nodes to transmit and receive signals simultaneously due to significant self-interference at the node, the cancellation of which is challenging and costly. Second, user throughput and packet delay performance rapidly deteriorates as the number of hops between the donor and users grows [11], [12]. Moreover, UE mobility and traffic demand changes lead to imbalanced traffic across UL and DL transmissions.

Therefore, it is essential to provide efficient delay-aware traffic routing and user scheduling solutions for mmWave IAB networks to improve the user-centric performance and balance the load in the network subject to the half-duplex and interference constraints and dynamic UL and DL traffic. Specifically, our work focuses on a joint optimization of the frame pattern, routing, and user allocation in the form of efficient link scheduling that considers delay-sensitive and delay-tolerant user applications. Since the pattern is announced in advance, we operate with a centralized frame-based link scheduling solution in IAB networks under predicted traffic demands.

B. Related Works

Wireless multi-hop networks have been an active area of research for a few decades [13], [11]. However, due to specific technology limitations and application use cases, not all approaches can be applied to the IAB systems. For example, the complexity of the link scheduling problem in wireless multi-hop networks largely depends on the underlying topology and is NP-hard in general due to a large number of possible link combinations for scheduling. It is worth noting that routing and link scheduling problems are typically addressed separately in past works. For instance, state-of-the-art near-optimal delayed column generation method [14] for solving the link scheduling problem in flow-based wireless multi-hop networks has inspired several data-driven approaches [15], [16] to reduce the size of the link search space for faster and more stable learning of the optimal link scheduling strategy. The work in [17] offers an elegant semi-centralized framework for traffic routing in IAB systems that aims at minimizing end-to-end communication latency. However, it does not optimize user scheduling at IAB nodes and cannot provide delay guarantees.

To address the scheduling and routing problems jointly, backpressure routing [18] and the maximum weighted matching (MWM) iterative algorithm for instantaneous link scheduling [19] are widely employed. Even though MWM algorithms are highly attractive as approximate solutions to the NP-hard weighted link scheduling problem due to their simple concept, they require continuous buffer and channel state information updates in every slot, which produces enormous overhead if implemented in real-world systems. Moreover, these algorithms only optimize the throughput performance of the system without considering delay-aware metrics. In addition, their complexity is either polynomial in the number of nodes or, in the best case, linear in the product of the number of nodes and links, which rapidly grows with the increasing network size or the number of communication links.

The complexity issues associated with the backpressure and MWM algorithms in IAB systems are tackled by the adoption of data-driven methods [17], [20], [21]. Specifically, reinforcement learning (RL) attracts growing attention due to its ability to learn efficient policies via interaction with the environment when the traffic or channel statistics is unknown [22]. The work in [21] utilizes RL to learn the optimal scheduling policy in IAB networks. It integrates a frame-based traffic prediction module to minimize the feedback overhead for online learning. However, the learned policy is sub-optimal and outperforms the backpressure alternative only under some traffic regimes. Several recent works on IAB propose scalable semi-centralized and distributed link scheduling solutions [23], [24], [25], which naturally stem from the backpressure concept and, therefore, focus on the network throughput optimization rather than on the satisfaction of user demands.

C. Our Contribution

Our work addresses important gaps in the existing research on scheduling and routing in IAB systems. Specifically, the majority of past studies consider routing and user scheduling separately and/or disregard practical system considerations, e.g., half-duplex constraints. On the other hand, the works where these constraints are accounted for focus on the network utility rather than on delay-aware or user-centric metrics. On top of this, past known solutions such as widely-employed backpressure and MWM algorithms encounter implementation challenges because a global control action needs to be computed at every time step [18]. Therefore, our contributions in this paper can be summarized as follows.

- We develop a novel 3GPP-compliant optimization framework that accounts for the specific features of IAB networks with the goal to satisfy diverse user demands without any particular assumption on the traffic model. Specifically, we formulate a new delay-aware link scheduling problem as MDP that accounts for realistic half-duplex constraint, non-negligible interference, and UL and DL directions of communication.
- We propose a practical customized genetic algorithm (GA) that delivers desirable system performance in terms of service satisfaction and packet delay of delay-sensitive flows. It converges to the preferred performance region three times faster than the reference MCTS algorithm. Moreover, it improves the packet delay by 15% on average as compared to the baseline backpressure algorithm.
- We assess the performance of IAB systems under a wide range of traffic load, deployment, and topology configurations. Based on these observations, we formulate practical recommendations for link scheduling in IAB networks subject to a given system setup.

The rest of this paper is organized as follows. Section II describes the system model, while Section III introduces the

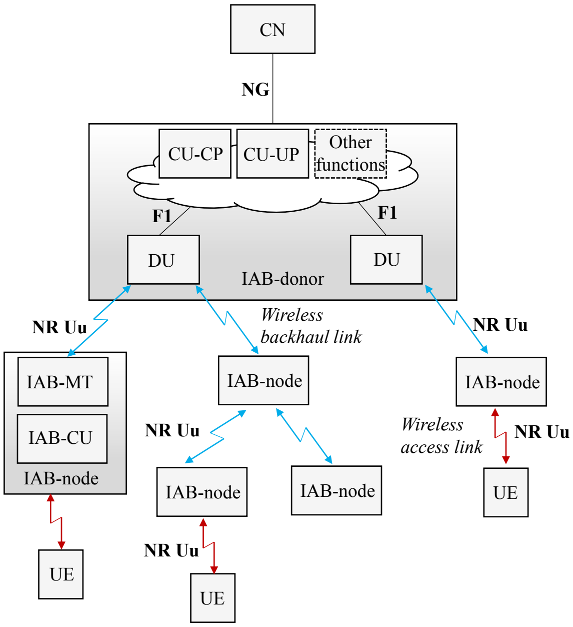


Fig. 1: Example IAB network with multi-hop topology [1].

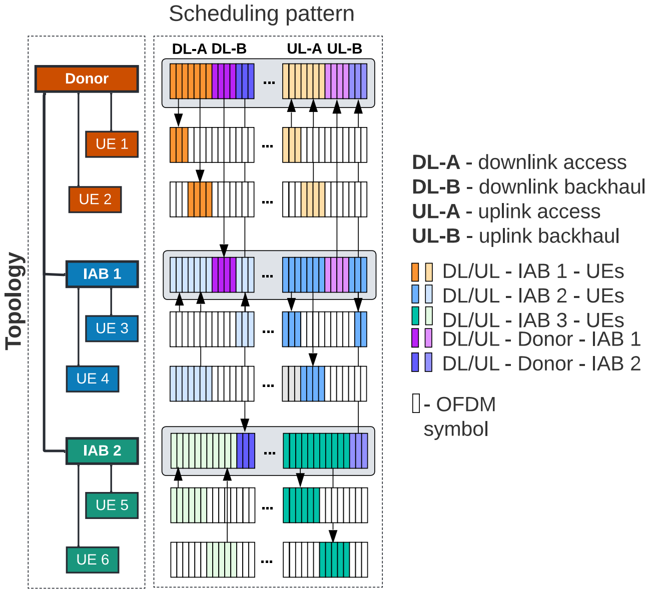


Fig. 2: Illustration of optimal scheduling and routing solution and a frame pattern allocation that can be derived from it.

problem formulation. Section IV provides details on the RL framework used to overcome the complexity of a given MDP, while Section V outlines the employed approximate solutions. Essential simulation assumptions and numerical results are discussed in Section VI. Finally, Section VII summarizes this work and discusses its potential extensions.

II. System Model

This section outlines the assumptions adopted for modeling a mmWave IAB network. We start by describing network deployment and user traffic dynamics. This is followed by a summary of the channel and antenna modeling procedure. Finally, the interference calculations are explained.

A. Topology, Routing, and User Demand Assumptions

We consider an IAB network with a single donor, V IAB nodes, and U UEs. The topology is assumed to be given by a graph $\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{0, \ldots, V\}$ represents the set of IAB nodes including the donor and $\mathcal{E} = \{(e_{ij})_{i,j \in \mathcal{V}}\}$ denotes the backhaul links. We consider two types of topologies, namely, DAG and spanning tree, which can be obtained as explained, e.g., in [26]. In spanning tree topology, each IAB node except the donor can have only one parent, while in DAG topology, IAB nodes can have up to V_p parents.

Each UE can generate data flows in UL and DL directions. To address delay requirements of different applications, we assume that UL and DL flows can be categorized into distinct classes of flows based on their sensitivity to packet delay. Without loss of generality, we consider two classes of flows, namely, delay-sensitive and delay-tolerant flows. Let F be the total number of flows. $\mathcal{F}_1$ denotes the set of flows with frame-based delay requirements, while $\mathcal{F}_0$ denotes the set of flows without any specific delay constraints. We let parameter δ control the ratio of delay-sensitive and delay-tolerant flows in the system. The classes are assigned randomly, such that $|\mathcal{F}_1| = \lfloor \delta F \rfloor$, while[1] $|\mathcal{F}_0| = \lceil (1-\delta)F \rceil$.

Each flow is associated with its source and destination nodes s_f and d_f from the joint set of UEs $\mathcal{U} = \{1, \ldots, U\}$ and donor node with index 0. Given the network topology $\mathcal{T}$, the number of paths between the source and the destination nodes can be more than one. We let $\mathcal{K}_n$ denote the set of flows, which have node $n \in \mathcal{N}$ in their paths. When the number of paths from s_f to d_f is higher than one, the routing decision for flow f is made dynamically by the nodes depending on the queue backlogs.

Demand r_f^* of flow $f \in \mathcal{F}$ is given as the number of packets to be delivered in a given frame to satisfy the timely throughput requirements. We assume that flow demands $(r_1^*, \ldots, r_F^*)$ and packet arrivals are known to the controller at the beginning of a frame and can be computed based on the *predicted* traffic load and perceived packet flow rates.

The goal is to find a scheduling and routing strategy that fulfills the flow demands and reduces the delay of delay-sensitive flows. As explained in Fig. 2, this strategy produces a *scheduling pattern*, which sets the order of link activations in space and time.

B. User Deployment and Communication Model

1) Propagation: We consider the urban macrocell (UMa) channel model as suggested for IAB networks in [1]. Accordingly, UEs are uniformly deployed over the area of interest, while the UE heights are uniformly distributed within interval $[1.5, 20]$ m.

Let x and y be the 2D and 3D distances between a UE and an IAB node. A UE may fall into either line-of-sight

[1] $\lfloor . \rfloor$ and $\lceil . \rceil$ represent flooring and ceiling operators, respectively.

$$\mathbb{P}_{\text{LoS}}(x) = \begin{cases} 1, & x \leq 18 \text{ m}, \\ \Big[\frac{18}{x} + \exp{(-\frac{x}{63})}(1-\frac{18}{x})\Big]\Big(1 + C^{'}(h)\frac{5}{4}(\frac{x}{100})^3 \exp{(-\frac{x}{150})}\Big), & 18 \text{ m} < x. \end{cases} \tag{1}$$

$$L_{dB}(x) = \begin{cases} 28 + 22\log_{10}(x) + 20\log_{10}(\omega_c), & \text{LoS}, 10 \text{ m} \leq x \leq d_{BP}, \\ 28 + 40\log_{10}(x) + 20\log_{10}(\omega_c) - 9\Big[\log_{10}(d_{BP})^2 - (h_{\text{BS}} - h)^2\Big], & \text{LoS}, d_{BP} \leq x \leq 5\text{km}, \\ 32.4 + 30\text{log}_{10}(x) + 20\log_{10}(\omega_c), & \text{NLoS}. \end{cases} \tag{2}$$

TABLE I: Table of notations.

Parameter	**Definition**
Network model	
F	Total number of flows
$\mathcal{F}$, $\mathcal{F}_1$, $\mathcal{F}_0$	Set of all flows, sets of delay-sensitive and delay-tolerant flows
r_f^*	Demand requirements for flow f
r_π^f	Achieved transmission rate of flow f under strategy π
δ	Fraction of delay-sensitive flows
$\mathcal{V}$, V	Set and number of IAB nodes and donor
$\mathcal{U}$, U	Set and number of UEs
$\mathcal{N}$, N	Set and total number of nodes in the system
T	Frame duration
$\mathcal{T} = \{\mathcal{V}, \mathcal{E}\}$	Network topology
$Q_n^f(t)$	Queue state of flow f at node n in time slot t
s_f, d_f	Source and destination nodes of flow f
$a_{ij}^f(t), \mathbf{a}(t)$	Link scheduling decision for flow f over link ij
$\mathcal{A}$	Set of all feasible schedules
π	Link scheduling strategy
$C_{ij}(t)$	Link capacity in number of packets
$b_{ij}^f(t)$	Transmission rate of flow f over link ij
N_b	Maximum number of simultaneously active beams at donor or IAB nodes
Communication model	
$L_{dB}(x), L(x)$	Propagation loss in dB and linear scales
ω_c	Carrier frequency in GHz
x, y	2D and 3D communication distances
d_{BP}	Breakpoint distance
h_{tx}, h_{rx}	Heights of TX and RX antennas
c	Speed of light
G_{ij}, G_{ij}^l	Antenna gain in the main communication direction in dBi and linear scales
W	Bandwidth
$P_{R_{ij}}$	Receive power at node j from node i
$P_{T_{ij}}$	Transmit power of node i toward node j
N_0	Noise power
$\Gamma_{ij}(x,t)$	SINR of the link between nodes i and j
$I_j(\mathbf{a}(t))$	Interference at node j for a given schedule $\mathbf{a}(t)$
$A_{ij}(\phi, \theta)$	Antenna radiation pattern from node i toward node j
ϕ, θ	Horizontal and vertical angular shifts
$A_{ij}^E(\phi, \theta)$	Radiation pattern of a single antenna element
w, v	Antenna side lobes weighting factor and phase shift
MDP model	
s_t, a_t	MDP state and action
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
$P(s'\|s,a)$	Transition probability
$R_\pi(s_0)$	Return of implementing strategy π in state s_0
$g_t(s_t, a_t)$	Immediate cost of taking action a_t in state s_t
$g_T(s_t)$	Terminal cost of being in state s_t

(LoS) or non-LoS (LoS) region, thereby experiencing different pathloss conditions [27]. Specifically, the LoS probability in (1) depends on the 2D distance x and UE height h, where

$$C^{'}(h) = \begin{cases} 0, & h \leq 13 \text{ m}, \\ \Big(\frac{(h-13)^{1.5}}{10}\Big), & 13 \text{ m} < h. \end{cases} \tag{3}$$

The path loss $L_{dB}(y)$ for the links in the LoS and NLoS conditions is provided by (2), where ω_c is the carrier frequency in GHz, h_{BS} denotes the height of an IAB node, and the break-point distance d_{BP} is given by

$$d_{BP} = 4\frac{(h-1)(h_{BS}-1)\omega_{c,Hz}}{c}, \tag{4}$$

where c denotes the speed of light and the carrier frequency $\omega_{c,Hz}$ is given in Hz.

For user association, we assume that each UE selects a node from $\mathcal{V}$ with the maximum reference signal received power (RSRP). Further, large-scale link fading is assumed to be known to the controller at the beginning of a frame and does not change during the frame duration.

2) Antenna and Interference: In our reference IAB deployment, the DgNB and IAB nodes comprise of three sectorized antennas with sufficient spatial separation to limit self-interference [28]. We assume that the beams in the transmit and receive directions of the intended communicating pair are perfectly aligned. Beyond that, we explicitly model antenna radiation patterns as recommended by [27] to evaluate interference, since the beams of interfering transmitters and intended receiver may overlap. An example of interference under a particular scheduling pattern is shown in Fig. 3. For each link in the scheduling pattern, all other links are treated as interfering.

According to our antenna model, the antenna radiation pattern is represented as a superposition of element radiation patterns. Therefore, the values of the half-power beamwidth (HPBW) and antenna gain depend on the number of antenna elements.

The radiation pattern $A(\phi_{ij}, \theta_{ij})$ of antenna at node i seen at node j is expressed by

$$\begin{aligned} A(\phi_{ij}, \theta_{ij}) =& A^E(\phi_{ij}, \theta_{ij}) + \\ & 10\log_{10}\Bigg(1 + \rho\Big[\sum_{m=1}^{N_H}\sum_{n=1}^{N_V}|w_{mn}v_{mn}|^2 - 1\Big]\Bigg), \end{aligned} \tag{5}$$

where $A^E(\phi_{ij}, \theta_{ij})$ stands for a single antenna element pattern, ϕ_{ij} and θ_{ij} are the horizontal and vertical angular shifts, respectively, w is the weighting factor responsible for the strength of side lobes, v is the phase shift, N_H and N_V are the numbers of antenna elements in horizontal and vertical planes,

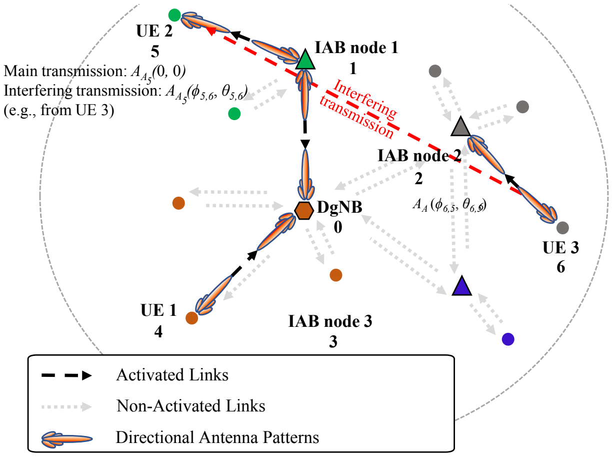

UE 2
5
IAB node 1
1
Main transmission: $A_{A_5}(0, 0)$
Interfering transmission: $A_{A_5}(\phi_{5,6}, \theta_{5,6})$
(e.g., from UE 3)
Interfering transmission
IAB node 2
2
$A_A(\phi_{6,5}, \theta_{6,5})$
DgNB
0
UE 3
6
UE 1
4
IAB node 3
3
Activated Links
Non-Activated Links
Directional Antenna Patterns

Fig. 3: Example interfering transmissions and simultaneously activated links at time slot t.

and ρ is the degree of correlation between the elements. The single antenna element pattern $A^E(\phi_{ij}, \theta_{ij})$ is computed as

$$A^E(\phi_{ij}, \theta_{ij}) = G_E - \min[-(A_{E_H}(\phi_{ij}) + A_{E_V}(\theta_{ij})), A_{max}], \tag{6}$$

where G_E is the gain of a single antenna element, A_{max} is the front to back ratio, while $A_{E_H}(\phi_{ij})$ and $A_{E_V}(\theta_{ij})$ are the attenuation values in horizontal and vertical planes, correspondingly [27], [29].

The gain in the main transmit direction of the intended communicating pair is

$$G^0_{ij} = A(0, 0). \tag{7}$$

We note that the antennas of the main transmitting–receiving pair are directed toward each other. Therefore, the beam misalignment angle equals zero, and the resultant gain in this direction is the best achievable, which is computed via (7). For instance, Fig. 3 shows an example link schedule, where UE 1 and UE 3 transmit toward the donor and IAB node 2, and IAB node 1 transmits toward UE 2 and the donor. As an example, let us consider the link between UE 2 and IAB node 1.

However, the horizontal and vertical beam misalignments ϕ_{kj} and θ_{kj} between the target receiver j and the interfering transmitter k are typically non-zero. For the example deployment in Fig. 3, the link from UE 3 to UE 2 is considered to be an interfering one. Having the coordinates of the transmitter (x_k, y_k, z_k) and the receiver (x_j, y_j, z_j), the direction of arrival can be computed as

$$(x_{kj}, y_{kj}, z_{kj}) = \begin{cases} \frac{(x_j - x_k)^2}{\sqrt{(x_j-x_k)^2+(y_j-y_k)^2+(z_j-z_k)^2)}} \\ \frac{(y_j - y_k)^2}{\sqrt{(x_j-x_k)^2+(y_j-y_k)^2+(z_j-z_k)^2)}} \\ \frac{(z_j - z_k)^2}{\sqrt{(x_j-x_k)^2+(y_j-y_k)^2+(z_j-z_k)^2)}}. \end{cases} \tag{8}$$

The corresponding angular shifts ϕ_{kj} and θ_{kj} can be expressed by converting the Cartesian coordinates in (8) into the spherical ones. Further, these values are utilized for weighting the incoming signals according to their directions of arrival and departure. Therefore, the antenna gain in the interfering direction for a receiving antenna is given by

$$G_{kj} = A(\phi_{kj}, \theta_{kj}). \tag{9}$$

A similar procedure is performed for the computation of the gain for a transmitted signal.

III. Problem Formulation

We assume that the IAB network operates in slotted time and formulate a scheduling and routing problem for the duration of a frame T. Let $Q^f_n(t)$ be the number of packets of flow f queued at node $n \in \mathcal{N}_f$ at time t. The network of interest has $\sum_{n \in \mathcal{N}} |\mathcal{K}_n|$ keep total queue length of different flows across all nodes.

Let $a^f_{ij}(t) \in \{0, 1\}$ be a scheduling decision representing whether the packets of flow f are transmitted over the link from node i to node j at time t. The joint scheduling decision produces a *scheduling pattern* $\mathbf{a}(t) = (a^f_{ij}(t))_{i,j \in \mathcal{N}, f \in \mathcal{F}}$. Due to the half-duplex constraint, a node cannot transmit and receive packets at the same time, whereas it can receive from or transmit to several nodes. Therefore, the pattern $\mathbf{a}(t)$ should comply with the following half-duplex constraint:

$$\sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{P}(n)} a^f_{in}(t) \cdot \sum_{f \in \mathcal{F}} \sum_{j \in \mathcal{P}(n)} a^f_{nj}(t) = 0, n \in \mathcal{V}, \tag{10}$$

where $\mathcal{P}(n)$ is a set of immediate neighbors to node n. Owing to the multi-beam transmission and reception capabilities at IAB nodes [19], the maximum numbers of simultaneously active outgoing and incoming transmissions are limited by N_b as follows:

$$\sum_{f \in \mathcal{F}} \sum_{i \in \mathcal{P}(n)} a^f_{in}(t) \leq N_b, n \in \mathcal{V}, \tag{11}$$

$$\sum_{f \in \mathcal{F}} \sum_{j \in \mathcal{P}(n)} a^f_{nj}(t) \leq N_b, n \in \mathcal{V}. \tag{12}$$

Scheduling pattern $\mathbf{a}(t)$ is *feasible* if constraints (10), (11), and (12) are met. We, thus, denote by $\mathcal{A}$ the set of all feasible scheduling patterns for a given network deployment.

Let $P_{T_{ij}}(\mathbf{a}(t))$ be the transmit power of node i to node j under the pattern $\mathbf{a}(t)$. If node $i \in \mathcal{N}$ transmits to node j the packets of only one flow, then it sends them with the maximum transmit power P_i. Otherwise, the power P_i is equally distributed across the flows yielding $P_{T_{ij}}(\mathbf{a}(t)) = P_i / \sum_{j \in \mathcal{N}} \sum_{f \in \mathcal{F}} a^f_{ij}(t)$.

Let $L(y)$ denote the pathloss (in the liner scale) between the nodes separated by distance y. As we demonstrate in subsection VI-B, for many scheduling patterns $\mathbf{a} \in \mathcal{A}$, the cross-link interference is non-negligible [30]. Therefore, the signal-to-interference-plus-noise ratio (SINR) over link (i, j) denoted by $\Gamma(y_{ij}, \mathbf{a}(t))$ is given as follows:

$$\Gamma(y_{ij}, \mathbf{a}(t)) = \frac{P_{T_{ij}}(\mathbf{a}(t)) G^0_{ij} G^0_{ji}}{(NW + I_j(\mathbf{a}(t))) L(y_{ij})}, \tag{13}$$

where N is the thermal noise power spectral density, W is the system bandwidth, G_{ij} and G_{ji} are the corresponding linear antenna gains at nodes i and j for the perfectly aligned beams, and $I_j(\mathbf{a}(t))$ stands for the interference power at the receiver j for a given pattern $\mathbf{a}(t)$. The latter can be obtained by

$$I_j(\mathbf{a}(t)) = \sum_{k \in \mathcal{K}_j(\mathbf{a}(t))} \frac{\sum_{m \in \mathcal{N}} a_{km}(t) P_{T_{km}} G_{kj} G_{jk}}{L(y_{kj})}, \quad (14)$$

where $\mathcal{K}_j(\mathbf{a}(t))$ is the set of interfering nodes to node j given the link scheduling pattern $\mathbf{a}(t)$, G is the antenna gain in the linear scale considering the corresponding horizontal and vertical angular shifts ϕ_{kj} and θ_{kj} for the current locations of the nodes. The interference can be computed independently for each node once the pattern $\mathbf{a}(t)$ is known.

The capacity of link (i, j) in slot t expressed as the number of packets is then

$$C_{ij}(\mathbf{a}(t)) = F_l\Big(W \log_2\Big(1 + \Gamma(y_{ij}, \mathbf{a}(t))\Big)\Big), \quad (15)$$

where F_l is a function that determines the number of transmitted packets for a chosen modulation and coding scheme [31].

We let $b_{ij}^f(\mathbf{a}(t))$ be the amount of data in flow f transmitted from node i to node j in slot t. The number of transmitted packets is determined by the capacity $C_{ij}(\mathbf{a}(t))$ and the number of packets in the backlog queue $Q_i^f(t)$, so that only the packets that are in the backlog queue can be transmitted subject to the capacity:

$$b_{ij}^f(\mathbf{a}(t)) = a_{ij}^f(t) \min[Q_i^f(t), C_{ij}(\mathbf{a}(t))]. \quad (16)$$

The numbers of packets in the queues as a result of the scheduling decision $\mathbf{a}(t)$ are given for all $f \in \mathcal{F}$ and $t \in \{0, \ldots, T-1\}$ by the following:

$$\begin{aligned} Q_n^f(t+1) =& Q_n^f(t) + \Lambda_n^f(t) + \\ & \sum_{i \in \mathcal{P}(n)} b_{in}^f(\mathbf{a}(t)) - \sum_{j \in \mathcal{P}(n)} b_{nj}^f(\mathbf{a}(t)), \end{aligned} \quad (17)$$

where $\Lambda_n^f(t)$ denotes the number of exogenously arrived packets and, therefore, $\Lambda_n^f(t) = 0$ for all IAB nodes $n \in \mathcal{V} \setminus \{0\}$ except for the donor. The initial system backlog $Q_n^f(0)$ for $n \in \mathcal{N} \setminus \{d_1, \ldots, d_F\}$ follows from the distribution Q_0, while $Q_{d_f}^f(0) = 0$ for $f \in \mathcal{F}$.

Let $r_f = Q_{d_f}^f(T)$ be the sizes of the destination queues for flows $f \in \mathcal{F}$ after T slots, which represent the numbers of delivered packets during a scheduling interval. The difference $r_f^* - r_f$ indicates whether a flow demand is satisfied. Consider a case where the initial flow backlog is greater than or equal to r_f^*. The demand of flow f is satisfied if $r_f^* - r_f \le 0$ and is assumed to be unfulfilled if $r_f^* - r_f > 0$ for $f \in \mathcal{F}$. It is also possible that $r_f^* - r_f < 0$ if all the packets in the backlog queues of flow f are delivered to the destination within T slots. We, therefore, require that $r_f^* - r_f$ is always greater than 0. If the initial flow backlog is less than r_f^*, the difference $r_f^* - r_f$ always exceeds 0. By minimizing $r_f^* - r_f$ over all flows, we aim at satisfying the flow demands, but cannot improve the delay of delay-sensitive flows $f \in \mathcal{F}_1$.

To reduce the latter, we introduce binary penalty $u_f(\mathbf{a}(t))$, $f \in \mathcal{F}_1$, to prioritize the scheduling of packets of delay-sensitive flows. It serves to impose a penalty if the size of the destination queue $Q_{d_f}^f(t)$ is not equal to the flow demand r_f^* of delay-sensitive flows $f \in \mathcal{F}_1$. The penalty in slot t equals 1 if $Q_{d_f}^f(t+1) \le r_f^*$ and equals 0 otherwise:

$$u_f(\mathbf{a}(t)) = \begin{cases} 0, & Q_{d_f}^f(t+1) \ge r_f^*, \\ 1, & Q_{d_f}^f(t+1) < r_f^*. \end{cases} \quad (18)$$

We let $\pi = (\mathbf{a}(0), \ldots, \mathbf{a}(T-1))$ be a centralized scheduling and routing strategy formulated as a sequence of scheduling decisions from the set Π of all feasible strategies.

Our goal is to find a strategy $\pi^* \in \Pi$ that minimizes the expected demand dissatisfaction $\max[r_f^* - r_f, 0]$ over all flows and improves the delay of delay-sensitive flows. This can be achieved by solving the following *optimization problem*:

$$\underset{\pi \in \Pi}{\text{Min:}}\ \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \sum_{f \in \mathcal{F}_0} u_f(\mathbf{a}(t)) + \sum_{f \in \mathcal{F}} \Big(\max[r_f^* - r_f, 0] \Big)^2 \right],$$
$$\text{Being subject to: (10), (11), (12), (16), (17), and (18).} \quad (19)$$

IV. Reinforcement Learning

The problem in (19) represents a finite horizon MDP. The size of Π in the worst case is $|\mathcal{A}|^T$. However, the number of practical scheduling patterns in every slot t, which avoid scheduling from empty queues, is usually significantly less than $|\mathcal{A}|$, but remains exponentially high. Therefore, we utilize a single-agent RL approach to find a close approximation for the solution of (19). In this section, we define RL-specific formulations that include states, actions, transition probabilities, and costs.

a) States: Let $s_t \triangleq (Q_i^f(t))_{i \in \mathcal{N}, f \in \mathcal{F}}$ be the state of the MDP at time t. All queues Q_n^f are finite and bounded by the demand r_f^* in a given frame. The state space $\mathcal{S}$ and its cardinality $|\mathcal{S}|$ can be given by

$$\mathcal{S} = \cup_{f=1}^F \mathcal{S}_f, \mathcal{S}_f = \{(Q_n^f)_{n \in \mathcal{N}_f} : Q_n^f = \{0, \ldots, r_f^*\}\}, \quad (20)$$

$$|\mathcal{S}| = (r_f^* + 1)^K. \quad (21)$$

b) Actions: At the beginning of $t \in \{0, \ldots, T-1\}$, the MDP agent chooses a *feasible* action $a_t \triangleq \mathbf{a}(t)$, $a_t \in \mathcal{A}$.

c) Transition probabilities: Since the arrivals $\Lambda_n^f(t)$ and large-scale fading are known to the MDP agent and do not change within a frame duration, any transition from state s_t to state s'_t after taking action a_t is deterministic. Hence, the transition probabilities $P(s'_t | s_t, a_t) = 1$ if state s'_t is produced via (16) and (17) from s_t by taking action a_t; otherwise, $P(s'_t | s_t, a_t) = 0$.

d) Costs: Let $g_t(s_t, a_t)$ and $g_T(s_T)$ be the cost of taking action a_t in state s_t and the cost of being in state s_T at the end of the problem horizon, respectively. The cost $g_t(s_t, a_t) = \sum_{f \in \mathcal{F}_0} u_f(a_t)$, while $u_f(a_t)$ is given by (18). The cost of being in state s_T after T slots $g_T(s_T) = \sum_{f \in \mathcal{F}} (\max[r_f^* - r_f, 0])^2$ represents demand dissatisfaction.

The *return* $R_\pi(s_0)$ represents the total cost obtained by following the strategy $\pi \in \Pi$ from state s_0 and is given by

$$R_\pi(s_0) = \mathbb{E}_\pi \Big[\sum_{t=0}^{T-1} g_t(s_t, a_t) + g_T(s_T) \Big]. \quad (22)$$

Algorithm 1 Q-Learning

```
Initialize Q-table, $\alpha$, $N_e$, $T$
for $n = 1, \dots, N_e$ do
    Reset the environment $s_0 \sim \mathcal{S}_0$
    for $t = 0, \dots, T-1$ do
        Sample action $a_t$ using $\varepsilon$-greedy policy
        Take action $a_t$ and observe $g(s_t, a_t)$, $s_{t+1}$
        if $s_t \notin$ Q-table then
            Add $q(s_t, a_t)$ to Q-table
        end if
        Update Q-table via (23)
    end for
end for
```

Minimization of $R_\pi(s_0)$ is equivalent to the problem in (19) subject to $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$. It is known as shortest path problem [32] and can be solved iteratively through the Bellman equation if the state and action spaces are small enough.

It is worth noting that for a given initial state s_0 there might be several paths with the same cost, i.e., the optimal strategy π is not unique. For deterministic problems, in contrast to stochastic ones, minimizing the cost over admissible actions a_t in every decision epoch results in the same optimal cost as minimizing over sequences of actions $\pi = (\mathbf{a}(0), \dots, \mathbf{a}(T-1))$, since the future states and control are determined via dynamic equations.

V. Approximate Solutions

In this section, we describe the algorithms, which can be adopted for solving the formulated MDP for a realistic network size and frame duration. We consider different algorithms to identify their advantages in relation to the addressed problem.

A. Q-Learning

First, we consider the Q-learning method, which uses a lookup table to find the best action in a given state. The so-called Q-table [33] specifies the value of an action taken in a particular state. The Q-table is initialized with zeros; then, the elements $q(s_t, a_t)$ of the table are iteratively updated as

$$q(s_{t+1}, a_{t+1}) = q(s_t, a_t) + \alpha(-g(s_t, a_t) + \max_a q(s_{t+1}, a) - q(s_t, a_t)), \tag{23}$$

where α is the learning rate. Note that we use negative reward $-g_t(s_t, a_t)$, which is defined in Section IV, since the original problem aims at minimizing the expected return. The solution is summarized in Algorithm 1, where N_e refers to the number of training episodes.

The learning of a Q-table is executed via the following steps. After the initialization of the table, the learning rate, and the number of training episodes, the initial state s_0 is drawn from a known distribution $\mathcal{S}_0 \in \mathcal{S}$. At every iteration, the actions are chosen randomly from $\mathcal{A}$ according to the ε-greedy policy. Specifically, action $a_t = \arg\max_a q(s_t, a)$ is selected with probability $1-\varepsilon$, while a random action from $\mathcal{A}$ is selected with probability ε. Then, after the corresponding cost is derived, the Q-table is updated by following (23). The algorithm runs for a fixed number of episodes, each of which episode is terminated after T steps. In our case, the terminal state corresponds to the queue states $\mathbf{Q}(T)$.

The time complexity of this method depends on the cardinalities of action and state spaces as $\mathcal{O}(T|\mathcal{A}||\mathcal{S}|)$. Moreover, buffer utilization under this algorithm increases over time as new actions and states are discovered. The use of deep Q-learning (DQL) helps tackle the problem of a growing Q-table. However, the process remains memory-heavy as it requires storing the states and actions in the replay buffer. As the learning agent may not encounter the majority of the states during the learning process, we further consider a Q-learning algorithm with prioritized sweeping that can significantly improve the performance of Q-learning in deterministic environments.

B. Q-Learning with Prioritized Sweeping

In conventional Q-learning, Q-values are updated in the order of agent experience, i.e., as they are encountered. In contrast, prioritized sweeping updates are based on the importance of the state–action pairs [34]. The respective steps are summarized in Algorithm 2.

We introduce P priorities for the encountered states and a priority queue PQ to store the most promising states. The priorities are computed via a temporal difference error between the discounted estimated value of the current state–action pair and the value of the next state-action pair added to the reward received. However, even though the priorities are updated for every state–action pair, only those that are above the threshold θ are stored in the priority queue PQ. The next state and reward of each state–action pair above the priority threshold are stored in the $Model$ array. During the Q-table update process, the last state–action pair from the priority queue is used for an update. Prioritized sweeping allows for a more directed search over the problem search space as it calculates the impact of a new state–action pair on all its predecessors and keeps track of only the important ones.

The theoretical complexity of the prioritized sweeping scheme is as high as that of the conventional Q-learning method. However, several studies, such as the one in [35], demonstrated empirically that the enhanced algorithm converges faster as compared to the conventional option due to its prioritized search.

C. Monte Carlo Tree Search

The MCTS [36] scheme seeks the best strategy by combining the tree search method and the sampling technique [37] to build a decision tree from the initial state. In this algorithm, the problem is represented via a *graph* where states are graph nodes. The initial state S_0 is named the root node, while a node extending from the root or another node is named a child node.

This method has become a state-of-the-art technique for deterministic combinatorial games and problems [38]. The

Algorithm 2 Q-Learning with Prioritized Sweeping

```
Initialize $q(s,a)$, $Model(s,a)$, $PQ$, $\theta$,$\forall s \in \mathcal{S}$, $\forall a \in \mathcal{A}(s)$
for $n = 1, \dots, N_e$ do
    $s_t \leftarrow$ current non-terminal state
    Sample action $a_t$ using $\varepsilon$-greedy policy
    Take action $a_t$, observe reward $g(s_t, a_t)$ and state $s_{t+1}$
    $Model(s_t, a_t) \leftarrow g(s_t, a_t), s_{t+1}$
    $P \leftarrow |g(s_t, a_t) + \gamma \max_a q(s_{t+1}, a) - q(s_t, a_t)|$
    if $P > \theta$ then insert $s_t, a_t$ into $PQ$ with priority $P$
    while $PQ$ is not empty do
        $s_t, a_t \leftarrow first(PQueue)$
        $g(s_t, a_t), s_{t+1} \leftarrow Model(s_t, a_t)$
        Update elements of Q-table via (23)
        for $\forall \bar{s}, \bar{a}$ predicted to yield $s_t$: do
            $\bar{g}(s_t, a_t) \leftarrow$ predicted reward for $\bar{s}, \bar{a}, s_t$
            $P \leftarrow |\bar{g}(s_t, a_t) + \gamma \max_a q(s_t, a) - q(\bar{s}, \bar{a})|$
            if $P > \theta$ then insert $\bar{a}$ into $PQ$ with priority $P$
        end for
    end while
end for
```

Algorithm 3 Customized Genetic Algorithm

```
Initialize population of size $N_{pop}$ randomly
Evaluate initial population
for $n = 1, \dots, N_e$ do
    for $i = 1, \dots, N_{pop}/2$ do
        Pick two parents with the best fitness score
        Reproduce
        Mutate
    end for
    for $j = N_{pop}/2, \dots, N_{pop}$ do
        Pick an individual $P_b$ with the best return $R_\pi$
        Identify flows with satisfied demands $\mathcal{F}_s$ in $P_b$
        Exclude $\mathbf{a} \in \mathcal{F}_s$ from action space $\mathcal{A}$
        for action $a(t)$ in $P_b$ do
            if $a(t) \in \mathcal{A}$ or satisfied flow queue is empty then
                Compare queues of unsatisfied flows
                Change $a(t)$
            end if
        end for
    end for
    Evaluate population
end for
```

fundamental challenge of balancing exploration and exploitation in MCTS is addressed in the same way as in multi-armed bandit (MAB) problems. The algorithm treats each state of the search tree as a MAB and selects an action that maximizes the upper confidence bound (UCB) heuristics. In particular, the algorithm consists of four main steps:

- **Selection.** At the initial step, all graph weights $\hat{v}$ are initialized as infinite. Therefore, a child node in a graph is selected randomly. Then, the child node is chosen based on the maximization of the UCB score, which is given by
$$\max_a \left(\frac{\sum_{k=1}^{N_v(s_t,a)} (-R_\pi)}{N_v(s_t, a)} + C_{UCB} \sqrt{\frac{\log(N_v(s_0))}{N_v(s_t, a)}} \right), \tag{24}$$
where $N_v(s_t, a)$ is the number of times action a has been selected in state s_t, while $N_v(s_0)$ is the total number of visits, $\sum_{k=1}^{N_v}(-R_\pi)$ is the reward accumulated over $N_v(s_t, a)$ when action a has been selected in state s_t, and c_{UCB} is a parameter responsible for the exploration–exploitation trade-off.
- **Expansion.** The search tree is expanded via all possible actions by adding child nodes to the leaf nodes.
- **Roll-out.** From a selected child node, the sequence of actions is chosen randomly at each depth of the tree until the terminal state is reached. Then, for this sequence of actions, an intermediate return is conducted to estimate the performance of the selected child node.
- **Backpropagation.** The reward returned from the previous step is backpropagated all the way up to every node by updating the accumulated rewards, the numbers of visits, and the corresponding UCB values.

These steps are repeated as many times as time or computational resources allow. A strategy is formed either after a fixed number of iterations or when a computational limit is reached.

The complexity behind a single update of the MCTS algorithm is $\mathcal{O}(T|\mathcal{A}|)$. On the other hand, the overall complexity of the method depends on the total number of iterations. In turn, the number of iterations is a function of multiple factors, such as, e.g., the initial state and deployment parameters.

D. Genetic Algorithm

GA represents a policy-based approach inspired by the biological evolution process [39]. It starts with an initial population, where an individual represents a scheduling policy $\pi = (\mathbf{a}(0), \dots, \mathbf{a}(T-1))$ for all $\mathbf{a}(t) \in \mathcal{A}$, while the gene of an individual represents a single pattern $a(t)$ for the time slot t. The fitness score of an individual, thus, corresponds to the expected return $R_\pi(s_0)$ starting from state s_0 and following policy π. The size of the population N_{pop} is chosen at the initialization step and does not change. Therefore, the algorithm complexity depends on this parameter as $\mathcal{O}(TN_{pop})$. Further, the memory utilization of GA is more efficient compared to Q-learning because it depends only on the population size N_{pop}, which remains unchanged throughout the simulation time.

The initial population is generated randomly, and the fitness score for every individual is then evaluated. The fitness score of the entire population is equivalent to the best fitness score among its individuals. A new population is obtained from the current one via the following steps:

- **Selection.** At the selection step, N_p individuals with the best fitness score are tagged within the population. These individuals are named parents, while others are named children. Further, two parents with the best fitness score are selected from the current population to reproduce at the next step.
- **Reproduction.** At the reproduction step, crossover is performed for all pairs of parents to produce offspring. The

crossover procedure is executed by selecting a crossover point within the parent sequences and by exchanging their genes beyond that point. The crossover point t of a parent is selected randomly from $[0, \dots, T-1]$.

- **Mutation.** Offspring individuals can be subject to a mutation, when pattern a_t at a randomly selected position of policy π is to be changed. Note that the newly produced individuals always account for the half-duplex constraint, because $\mathbf{a}(t)$ is selected from the set of actions $\mathcal{A}$.

Both crossover and mutation procedures are performed with certain probabilities P_c and P_m, which are set at the initial step of the algorithm execution. The above steps are repeated until the time/computational limit or a given number of iterations is reached.

E. Customized Genetic Algorithm

Even though GA demonstrates adequate operation in combinatorial search problems, its performance may drop due to the random population initialization, selection, crossover, and mutation steps [40], [41]. To overcome this shortcoming, we develop a customized version of the GA for our problem. The pseudo-code of our customized GA is summarized in Algorithm 3.

In the modified GA, the population is generated via two different methods. Specifically, the first half of the population follows the rules of the classical GA method, while the other half is produced via a customized approach. At the same time, the best individual is determined at each step. The second half represents a modification of the best individual (scheduling policy) being obtained by executing lines 10-19 in Algorithm 3. It is worth noting that such a population split offers more diversity and decreases the probability of falling into a local minimum.

The complexity of the modified GA version is similar to that of the classical implementation in the worst-case scenario. However, the customized algorithm may converge faster than the basic GA for the problem of interest. This is because the number of iterations that the GA needs to converge is subject to random mutations and crossovers, which may not guarantee reasonable convergence. On the contrary, our customized GA has more control over how the scheduling patterns are updated, because it aims to improve the current schedule as long as possible rather than update the links in a pattern randomly.

The links within the scheduling pattern a_t which can be changed are selected based on the buffer states. Specifically, for the current best policy, the number of transmitted packets is compared to the target requirements r_f^*. Those links, which contain flows where the requirements were satisfied are considered to be fixed, while all other links can be changed. Note that it is also necessary to compare not only the requirements but also the buffer states of these flows to avoid scheduling a packet transmission from an empty buffer. After identifying the links that can be modified, we reduce the action space by excluding those actions, which involve only 'satisfied' flows. Moreover, before an action change, we track the queues of 'unsatisfied' flows to determine the number of potential changes in policy π needed to satisfy the requirements as well as which flows should be prioritized for the change. In the following section, we present a comparative assessment of the discussed algorithms.

TABLE II: Parameters utilized in numerical assessment.

Parameter	Value
Deployment parameters	
Carrier frequency, ω_c	30 GHz
Bandwidth, W	400 MHz
Cell radius	500 m
Tx power of donor, P_T	40 dBm
Tx power of IAB, P_T	33 dBm
Tx power of UE, P_T	23 dBm
Noise figure of donor and IAB node, f_n	7 dB
Noise figure of UE, f_n	13 dB
Power spectral density of noise, N_0	-173.93 dBm/Hz
Antenna array size of UE, $N_H \times N_V$	4x4
Antenna array size of DgNB and IAB node, $N_H \times N_V$	16x16
Height of DgNB	25 m
Height of IAB node	10 m
Height of UE	1.5 m
LoS fading variance, σ^2_{LoS}	4 dB
NLoS fading variance, σ^2_{NLoS}	7.8 dB
Gain of a single antenna element, G_E	8 dBi
Front to back antenna ratio, A_{max}	30 dB
Algorithmic parameters	
Learning rate, α	0.1
Maximum number of episodes for Q-learning	50000
Population size	1000

VI. Numerical Results

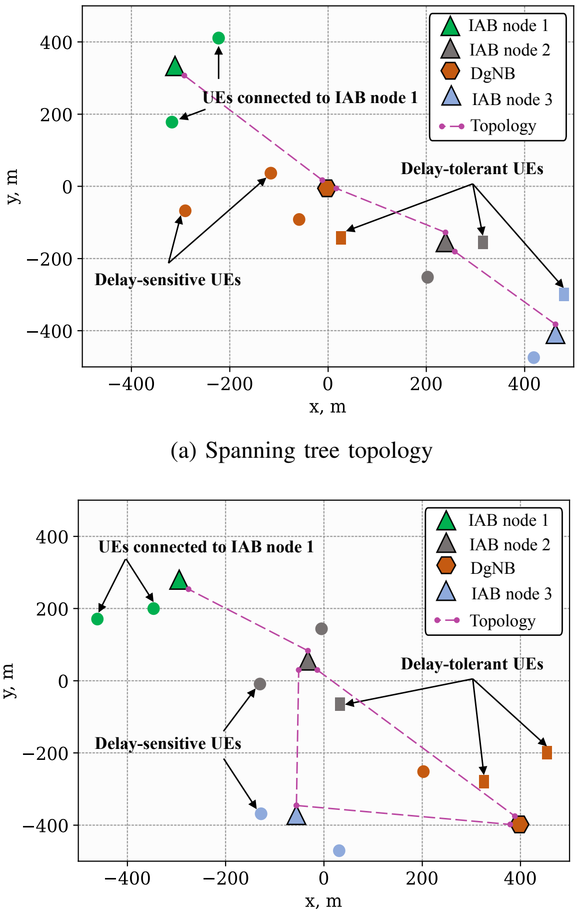
A. Simulation Assumptions

We consider a large number of IAB network deployments, which results in many different realizations of spanning tree and DAG topologies. In particular, the DgNB is placed at the center or at the edge of the cell, while UEs are uniformly distributed within the cell, and IAB nodes are positioned within the cell by ensuring sufficient distance between each other and the DgNB as recommended in [1]. Examples of the IAB deployments with realistic tree and DAG topologies are given in Fig. 4a and Fig. 4b, respectively.

We utilize our custom simulation software written in Python programming language to assess the performance of the considered strategies [42]. The modeling parameters are 3GPP-compliant and can be found in Table II. The parameters of the approximation algorithms are selected empirically, i.e., after evaluation of different values, the preferred ones are chosen.

We assume that for the selected numerology, the time slot Δt is 1 ms and it is represented by 14 OFDM symbols. This is a reasonable assumption due to the practical limitations of beamforming. However, one can assume that Δt is equal to the duration of an OFDM symbol or a sequence of OFDM symbols to conduct scheduling with a desired granularity.

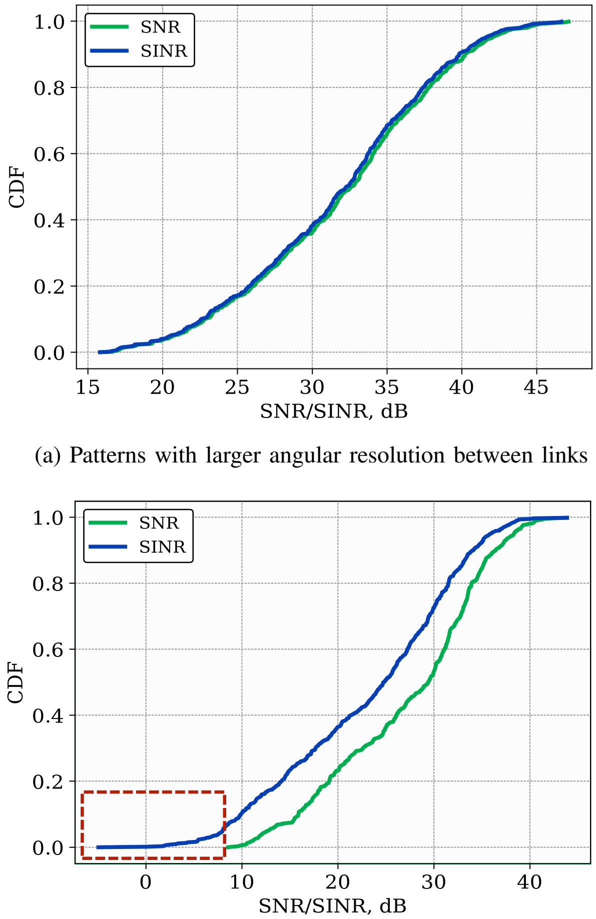
Without loss of generality, we assume that $\Lambda_n^f(0) = r_f^*$ and $\Lambda_n^f(t) = 0$ for $t \in [1, \dots, T-1]$ and $f \in \mathcal{F}$. The flow demand r_f^* can either be obtained from real data or sampled as follows. In order to understand the impact of heterogeneous demands on the system performance, we introduce parameter $\sigma \in \{0.1, \dots, 0.9, 1\}$ that captures how far the flow demands are from the capacity region [19] for a given topology and



(a) Spanning tree topology

(b) DAG topology

Fig. 4: Examples of IAB network deployments.



(a) Patterns with larger angular resolution between links

(b) Patterns with smaller angular resolution between links

Fig. 5: SNR/SINR CDF for different scheduling patterns.

number of users. Here, $\sigma = 0$ means that all the demands are within the network capacity region and $\sigma = 1$ means that each demand exceeds the capacity. For every value of σ, we generate different combinations of flow demands and then average the results of simulation runs collected for each combination.

With respect to unseen topologies, the scheduling strategy is updated every frame as the initial state of the MDP S_0 changes. The latter means that the initial queue backlogs and channel states might be updated. However, all the considered algorithms utilize previously learned statistics and strategy, which facilitates finding a new strategy. Whenever the network deployment changes, one needs to update state space $\mathcal{S}$ and action space $\mathcal{A}$ with respect to a new number of IAB nodes, UEs, and network topology before solving the target problem again.

B. Intra-Cell Interference and Small-Scale Fading

In this subsection, we verify our assumption on the presence of non-negligible interference under certain scheduling patterns. Specifically, we expect to see only marginal interference when the distances between all active transmitters are large enough, and their antenna radiation patterns do not overlap significantly.

To demonstrate this effect, we compute the SINR across the transmission links for all possible scheduling patterns in a given deployment and average the results for two representative examples. In the first set of patterns, the transmitters are distributed sparsely, and their beams aim in different directions. In the second set of patterns, the transmitters are located nearby, such that the angular resolution between the main and interfering transmissions is less than 20-25°. It was previously demonstrated in [28] that such an angular resolution or higher is sufficient to guarantee negligible interference. Fig. 5a and Fig. 5b illustrate the CDFs of the signal-to-noise ratio (SNR) and SINR for each of the two sets of scheduling patterns. To obtain these CDFs, we consider small-scale fading $\ln(X_\sigma) \sim \mathcal{N}(0, \sigma^2)$, where σ^2 takes different values for LoS and NLoS links.

In Fig. 5a, the interference impact is negligible. However, the effect of interference becomes more noticeable for the second set of scheduling patterns as demonstrated in Fig. 5b. The average SINR is 3.4 dB lower as compared to SNR. Moreover, the tail of the distribution indicates that the link capacity can drastically decrease and SNR unlike SINR cannot

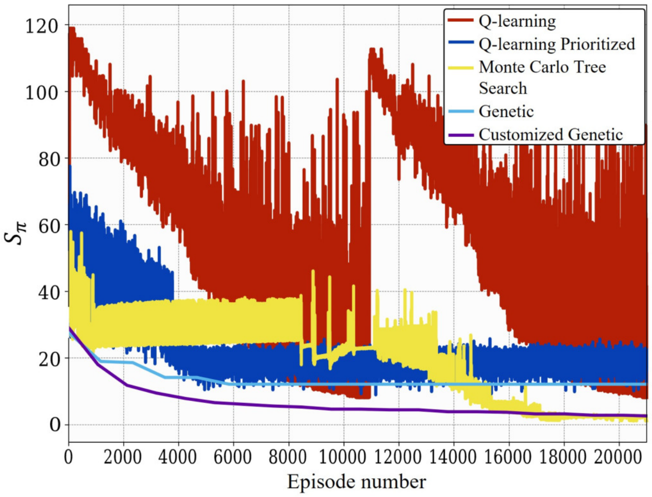


Fig. 6: Demand dissatisfaction rate.

capture this degradation. Despite the fact that the gap between SNR and SINR may not be significant on average, accounting for interference when selecting a scheduling pattern is essential because some patterns can cause link outage if, e.g., UEs are located close to a shared border of two adjacent antenna sectors [28].

Our observations suggest that the link capacity degradation is dependent on the link scheduling pattern rather than the small-scale channel variations.

C. Convergence Analysis

We compare the performance of conventional Q-learning, Q-learning with prioritized sweeping, MCTS, GA, and customized GA in terms of the *demand dissatisfaction rate*

$$S_\pi = \sum_{f \in \mathcal{F}} \max[r_f^* - r_f, 0]^2. \tag{25}$$

Due to the prohibitive complexity of the value iteration method for solving the MDP problem in (19) even for a small-scale deployment, the true optimal value $R_{\pi^*}(s_0)$ is unavailable. Therefore, comparing the values of $R_\pi(s_0)$ achieved by the above methods may not provide a clear understanding on the satisfaction of traffic demands. Moreover, the addition of binary penalty does not reflect how close the achieved rates are to the desired ones. On the contrary, demand dissatisfaction rate S_π has its minimum at 0. Note, however, that this value might generally be unreachable in deployments with arbitrary demands.

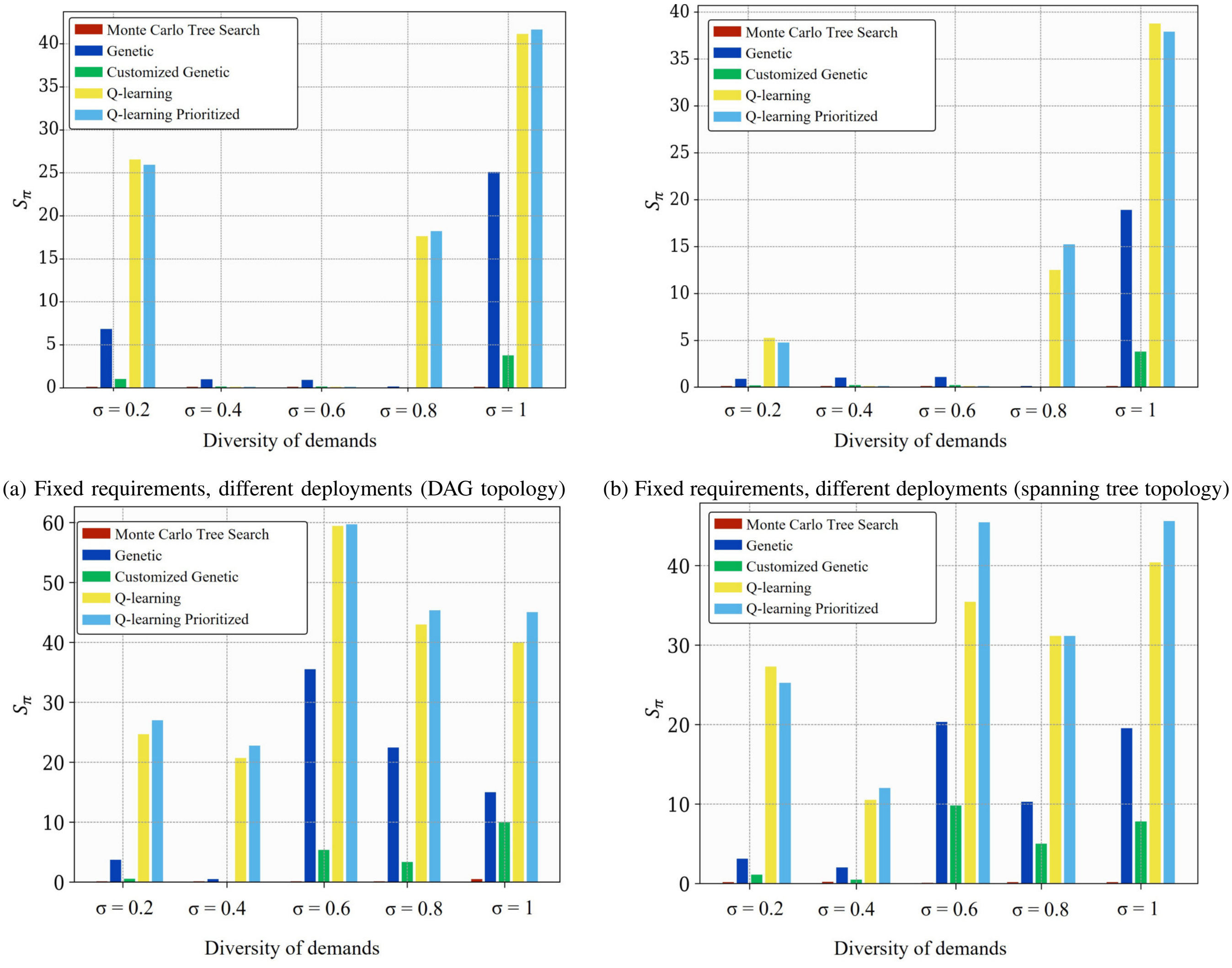
Fig. 6 compares the convergence of the baseline methods to the minimal demand dissatisfaction. For these results, we fix the flow demands and consider the scheduling interval T to be equal to 80 slots or one episode. The performance in time is compared in terms of iterations, where one iteration represents the choice of one scheduling pattern in a frame. Due to the fact that the conventional and customized GAs estimate $N_p = 1000$ link schedules (policies) per one iteration, the curves in Fig. 6 for these two methods are scaled correspondingly for a fair comparison in terms of time. The experiment spans over 20000 episodes for all other algorithms, which corresponds to $N_e = 250$ episodes of the conventional and customized GAs.

As can be seen in Fig. 6, the MCTS algorithm achieves the lowest demand dissatisfaction rate over 20000 iterations. However, its computational complexity is higher as compared to, e.g., GAs. On the other hand, we demonstrate that by customizing the basic GA, the performance can closely match that of MCTS while maintaining the computational efficiency of the conventional GA. Therefore, a sub-optimal solution can be achieved after a smaller number of iterations. Moreover, all other algorithms display better dissatisfaction rates until 14500 iterations, after which the values obtained with the help of MCTS rapidly converge to the minimum. As demonstrated in other applications [36]–[38], MCTS is subject to a long exploration period, which, on the other hand, facilitates finding the global minimum rather than stalling in the local ones. It is an appropriate benchmark when the true optimum is unknown but remains computationally heavy in practice, since the duration of the exploration period grows with the number of flows and nodes in the system.

The Q-learning and prioritized Q-learning options demonstrate the worst performance as these algorithms tend to converge to a local minimum. The prioritized Q-learning scheme performs better than the conventional Q-learning in terms of the convergence time, but it is even more prone to convergence to a local minimum. Moreover, both algorithms have memory utilization issues, which can be tackled by employing GAs. The proposed GA customization allows for faster convergence as compared to the basic GA, while inheriting its advantages in resource utilization. The latter makes it attractive for implementation in real-world systems. From the system design perspective, it means that the customized GA is more suitable for larger-scale deployments, whereas MCTS can be applied in smaller-scale networks.

D. Optimality Gap Analysis

Further, we consider the behavior of the optimality gap across a wide range of deployments and user traffic demands. Fig. 7a and Fig. 7b demonstrate the average demand dissatisfaction rate over 20 realizations of the DAG and spanning tree topologies with identical user traffic demands. In addition, the parameter σ represents demand diversity of the traffic flows. The purpose of introducing this parameter is to understand whether the performance of the considered algorithms depends on the traffic demands. In this experiment, deployment variations have varying link conditions, while UEs remain associated with the same parent node. Notably, the assumed variations in the link quality do not violate the maximum RSRP association rule as they mainly correspond to the variations in the link length within the radii of the parent IAB nodes or donor. In the second experiment, the link quality is fixed, but the user traffic demands are varied. Its results for the considered DAG and spanning tree topologies are given, respectively, in Fig. 7c and Fig. 7d. We remind that the figures obtained for a single value of σ are the average demand dissatisfaction rates observed over the deployments with different demands.



(a) Fixed requirements, different deployments (DAG topology)

(b) Fixed requirements, different deployments (spanning tree topology)

(c) Fixed deployments, different requirements (DAG topology)

(d) Fixed deployments, different requirements (spanning tree topology)

Fig. 7: Best demand dissatisfaction rate achieved by the considered algorithms.

As evident from the results of both experiments in Fig. 7, MCTS algorithm is the best solution for finding a close-to-optimal predictive schedule that satisfies all traffic demands. Specifically, it performs equally well for the deployments with arbitrarily DAG and tree topologies as well as with highly asymmetric traffic loads (where $\sigma = 1$). Note that one of the advantages of the MCTS scheme is that it can be terminated when any of the specified stopping conditions are met. Examples are computational resources or time budget, target or acceptable demand dissatisfaction rate. These conditions may be implemented specifically by taking into account the computing capabilities at the network controller or the deadline by which the predictive schedule needs to be constructed.

Q-learning-based algorithms demonstrate an inconsistency in their performance with respect to the changing demands. Even though Q-learning and Q-learning with prioritized sweeping are able to achieve close-to-optimal dissatisfaction rates in some setups, e.g., where $\sigma = 0.4$ as demonstrated in Fig. 7a, they fail to do so in the majority of other cases and demonstrate the worst results on average. The poor performance of the off-policy learning methods represented by Q-learning is likely due to the large action space. In general, this class of methods is more suitable for offline training scenarios where a pre-trained model is deployed in the real-world system and is expected to perform moderately well while keeping the learning of a better schedule in an online manner. However, this does not apply to link scheduling in IAB systems, where the use of Q-learning or DQL models requires slot-by-slot feedback aggregation from the IAB nodes to compute a link scheduling pattern for the upcoming time slot. A combination of these methods with system state prediction as demonstrated in [21] resolves the feedback issue but does not reduce the optimality gap. In particular, the performance gain of Q-learning-based solutions vanishes with the network scale growth and converges to the performance of the MWM algorithm [21] for link scheduling pattern optimization.

The demand dissatisfaction rate achieved by the GA and customized GA is smaller on average as compared to that of the Q-learning methods both with and without prioritized

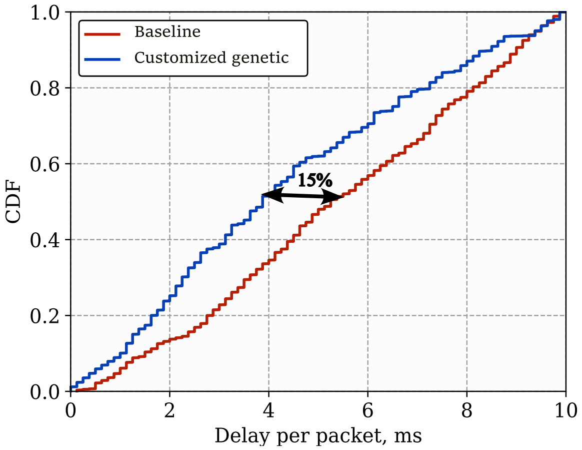


Fig. 8: Delay comparison for baseline vs. proposed customized GA.

sweeping. In all the experiments, the results shown by the customized GA are at least 2 times better than the performance of the baseline GA. It is more sensitive to highly dissimilar demands than MCTS and persistently demonstrates second-best performance. At the same time, the amount of resources and the complexity of necessary operations required by the customized GA are significantly smaller as compared to MCTS. Moreover, it converges faster to its best sub-optimal result than any other algorithm. Therefore, we conclude that our customized GA achieves an attractive performance–complexity trade-off.

E. Delay Performance

We now focus on the delay assessment by comparing the CDF of the packet delay achieved by using our customized GA variant with the CDF computed by using the reference predictive backpressure scheme [21] as the baseline. It is worth mentioning that the delay obtained with the MCTS method may be the smallest as it provides closer-to-optimal performance. However, the main drawback of this algorithm is in its long convergence time. Therefore, we consider the customized GA as it demonstrates a preferred performance–complexity trade-off.

The results are reported in Fig. 8. The proposed customized GA improves the packet delay by 15% on average as compared to the baseline. Such performance gains are mainly due to the penalty term introduced for the delay-sensitive flows by the objective function in (18). The cumulative penalty grows as long as the packets of delay-sensitive flows remain in the queues. Hence, our proposed algorithm tends to schedule delay-sensitive flows first whenever possible and address the residual dissatisfied demands of delay-tolerant flows later. Therefore, the customized GA can achieve near-optimal results in terms of the demand dissatisfaction rate fast and perform delay-aware scheduling that reduces the system packet delay.

VII. Conclusions

We develop a practical centralized delay-aware link scheduling solution for IAB networks with dynamic user demands. The proposed scheme takes into account realistic half-duplex constraints of IAB systems together with possible non-negligible cross-link interference to address transmission scheduling in both uplink and downlink directions over potentially multi-hop paths. The outlined framework allows for efficient handling of dissimilar UE demands and accounts for different delay requirements. Moreover, the developed solution helps balance the load across IAB nodes by dynamically selecting the next hop if multiple paths to the destination are available.

To address the high complexity of the formulated MDP, we employ different RL algorithms, including Q-learning, MCTS, and GAs. Our proposed customized GA method demonstrates the best performance–complexity trade-off among the reference solutions. It also achieves a significant 15% packet delay reduction across the considered types of traffic flows and various system configurations as compared to the backpressure algorithm. In addition, we provide system design recommendations based on our comparison of the alternative algorithms.

As future work, one may consider exploring a distributed optimization framework based on the problem formulation addressed in this paper to improve the scalability of the developed method. We aim at comparing centralized and distributed solutions in terms of scalability, convergence, and incurred overheads.

Acknowledgement

This work was supported by Intel Corporation and by the Research Council of Finland (Projects RADIANT, ECO-NEWS, SOLID, and ALL-ON).

References

[1] 3GPP, “Study on integrated access and backhaul (Release 16),” *3GPP TR 38.874 V16.0.0*, 2018.

[2] M. Cudak, A. Ghosh, A. Ghosh, and J. Andrews, “Integrated access and backhaul: A key enabler for 5G millimeter-wave deployments,” *IEEE Communications Magazine*, vol. 59, no. 4, pp. 88–94, 2021. DOI: 10.1109/MCOM.001.2000690.

[3] Y. Jiang, Y. Zhong, and X. Ge, “IIoT data sharing based on blockchain: A multileader multifollower Stackelberg game approach,” *IEEE Internet of Things Journal*, vol. 9, no. 6, pp. 4396–4410, 2021.

[4] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter wave cellular wireless networks: Potentials and challenges,” *Proceedings of the IEEE*, vol. 102, no. 3, pp. 366–385, 2014.

[5] 3GPP, “NG-RAN; Architecture description (Release 16),” 3GPP TS 38.401, 2022.

[6] S. Niknam, A. Roy, H. S. Dhillon, *et al.*, “Intelligent O-RAN for beyond 5G and 6G wireless networks,” in *2022 IEEE Globecom Workshops (GC Wkshps)*, 2022, pp. 215–220.

[7] M. Abbasi, A. Shahraki, and A. Taherkordi, “Deep learning for network traffic monitoring and analysis (NTMA): A survey,” *Computer Communications*, vol. 170, pp. 19–41, 2021.

[8] D. A. Tedjopurnomo, Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin, "A survey on modern deep neural network for traffic prediction: Trends, methods and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 4, pp. 1544–1561, 2022. DOI: 10.1109/TKDE.2020.3001195.

[9] O. Teyeb, A. Muhammad, G. Mildh, E. Dahlman, F. Barac, and B. Makki, "Integrated access backhauled networks," in *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, 2019, pp. 1–5.

[10] T. K. Vu, M. Bennis, S. Samarakoon, M. Debbah, and M. Latva-aho, "Joint in-band backhauling and interference mitigation in 5G heterogeneous networks," in *European Wireless 2016; 22th European Wireless Conference*, 2016, pp. 1–6.

[11] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, 2000. DOI: 10.1109/18.825799.

[12] A. Zemlianov and G. de Veciana, "Capacity of ad hoc wireless networks with infrastructure support," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 3, pp. 657–667, 2005. DOI: 10.1109/JSAC.2004.842536.

[13] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, Association for Computing Machinery, 2003, 66—80. DOI: 10.1145/938985.938993. [Online]. Available: https://doi.org/10.1145/938985.938993.

[14] Y. Cheng, X. Cao, X. S. Shen, D. M. Shila, and H. Li, "A systematic study of the delayed column generation method for optimizing wireless networks," in *Proceedings of the 15th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Philadelphia, Pennsylvania, USA: Association for Computing Machinery, 2014, 23–32, ISBN: 9781450326209. DOI: 10.1145/2632951.2632978. [Online]. Available: https://doi.org/10.1145/2632951.2632978.

[15] L. Liu, B. Yin, S. Zhang, X. Cao, and Y. Cheng, "Deep learning meets wireless network optimization: Identify critical links," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 167–180, 2020. DOI: 10.1109/TNSE.2018.2827997.

[16] S. Zhang, B. Yin, W. Zhang, and Y. Cheng, "Topology aware deep learning for wireless network optimization," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 9791–9805, 2022.

[17] A. Ortiz, A. Asadi, G. H. Sim, D. Steinmetzer, and M. Hollick, "SCAROS: A Scalable and Robust Self-Backhauling Solution for Highly Dynamic Millimeter-Wave Networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 12, pp. 2685–2698, Dec. 2019, ISSN: 1558-0008. DOI: 10.1109/JSAC.2019.2947925.

[18] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," in *29th IEEE Conference on Decision and Control*, 1990, pp. 2130–2132.

[19] F. Gomez-Cuba and M. Zorzi, "Optimal link scheduling in millimeter wave multi-hop networks with MU-MIMO radios," en, *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 1839–1854, Mar. 2020, ISSN: 1536-1276, 1558-2248. DOI: 10.1109/TWC.2019.2959295. [Online]. Available: https://ieeexplore.ieee.org/document/8937042/ (visited on 11/07/2021).

[20] L. Hai, Q. Gao, J. Wang, H. Zhuang, and P. Wang, "Delay-Optimal Back-Pressure Routing Algorithm for Multihop Wireless Networks," en, *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2617–2630, Mar. 2018, ISSN: 0018-9545, 1939-9359. DOI: 10.1109/TVT.2017.2770183. [Online]. Available: https://ieeexplore.ieee.org/document/8097024/ (visited on 12/17/2021).

[21] M. Gupta, A. Rao, E. Visotsky, A. Ghosh, and J. G. Andrews, "Learning Link Schedules in Self-Backhauled Millimeter Wave Cellular Networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8024–8038, Dec. 2020, ISSN: 1558-2248. DOI: 10.1109/TWC.2020.3018955.

[22] D. Gan, X. Ge, and Q. Li, "An optimal transport-based federated reinforcement learning approach for resource allocation in cloud-edge collaborative IoT," *IEEE Internet of Things Journal*, vol. 11, no. 2, pp. 2407–2419, 2024.

[23] M. Pagin, T. Zugno, M. Polese, and M. Zorzi, "Resource management for 5G NR integrated access and backhaul: A semi-centralized approach," *IEEE Transactions on Wireless Communications*, vol. 21, no. 2, pp. 753–767, 2022. DOI: 10.1109/TWC.2021.3098967.

[24] S. Gopalam, S. V. Hanly, and P. Whiting, "Distributed and local scheduling algorithms for mmwave integrated access and backhaul," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1749–1764, 2022. DOI: 10.1109/TNET.2022.3154367.

[25] V. F. Monteiro, F. Lima, M Rafael, *et al.*, "Paving the way towards mobile IAB: Problems, solutions and challenges," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2347–2379, 2022.

[26] M. Simsek, O. Orhan, M. Nassar, O. Elibol, and H. Nikopour, "IAB topology design: A graph embedding and deep reinforcement learning approach," *IEEE Communications Letters*, vol. 25, no. 2, pp. 489–493, 2021. DOI: 10.1109/LCOMM.2020.3029513.

[27] 3GPP, "Study on channel model for frequencies from 0.5 to 100 GHz (Release 15)," *3GPP TR 138.901 V15.0.0*, 2018.

[28] Y. Sadovaya, D. Moltchanov, H. Nikopour, *et al.*, "Self-interference assessment and mitigation in 3GPP IAB deployments," in *IEEE International Conference on Communications*, 2021, pp. 1–6.

[29] 3GPP, "Study of AAS base station (Release 12)," *3GPP TR 37.840 V1.0.0*, 2012.

[30] R. Barazideh, O. Semiari, S. Niknam, and B. Natarajan, "Reinforcement learning for mitigating intermittent interference in terahertz communication networks," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2020, pp. 1–6.

[31] 3GPP, "Technical specification group radio access network; NR; physical layer procedures for data (Release 16)," *3GPP TS 38.214 V16.4.0*, 2020.

[32] D. Bertsekas, *Dynamic Programming and Optimal Control - v1*, 3rd. Athena Scientific, 2005, vol. 1.

[33] P. Auer, T. Jaksch, and R. Ortner, "Near-optimal regret bounds for reinforcement learning," en, *Advances in Neural Information Processing Systems*, vol. 21, p. 38, 2008.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT press, 2018.

[35] J. Brea, "Is prioritized sweeping the better episodic control?" *arXiv preprint arXiv:1711.06677*, 2017.

[36] S. Gelly and D. Silver, "Combining online and offline knowledge in UCT," en, in *Proceedings of the 24th international conference on Machine learning*, ACM Press, 2007, pp. 273–280, ISBN: 978-1-59593-793-3. DOI: 10.1145/1273496.1273531. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1273496.1273531 (visited on 06/27/2022).

[37] H. S. Chang, M. C. Fu, J. Hu, and S. I. Marcus, "An Adaptive Sampling Algorithm for Solving Markov Decision Processes," en, *Operations Research*, vol. 53, no. 1, pp. 126–139, Feb. 2005, ISSN: 0030-364X, 1526-5463. DOI: 10.1287/opre.1040.0145. [Online]. Available: http://pubsonline.informs.org/doi/10.1287/opre.1040.0145 (visited on 06/27/2022).

[38] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte Carlo tree search: A review of recent modifications and applications," *Artificial Intelligence Review*, pp. 1–66, 2022.

[39] O. Kramer, "Genetic algorithms," in *Genetic algorithm essentials*, Springer, 2017, pp. 11–19.

[40] C. Madapatha, B. Makki, A. Muhammad, E. Dahlman, M.-S. Alouini, and T. Svensson, "On Topology Optimization and Routing in Integrated Access and Backhaul Networks: A Genetic Algorithm-Based Approach," en, *IEEE Open Journal of the Communications Society*, vol. 2, pp. 2273–2291, 2021, ISSN: 2644-125X. DOI: 10.1109/OJCOMS.2021.3114669. [Online]. Available: https://ieeexplore.ieee.org/document/9548327/ (visited on 06/08/2022).

[41] T. Wang, G. Zhang, X. Yang, and A. Vajdi, "Genetic algorithm for energy-efficient clustering and routing in wireless sensor networks," *Journal of Systems and Software*, vol. 146, pp. 196–214, 2018.

[42] *IAB deployment modeling profile*. [Online]. Available: https://github.com/yekaterina-sadovaya/IAB-Sim.

Yekaterina Sadovaya is a Doctoral Candidate at Tampere University, Finland. She received her M.Sc. in Information Technology from Tampere University, Finland, and from Peter the Great Saint Petersburg Polytechnic University, Saint Petersburg, Russia, in 2021 and 2020, and her B.Sc. degree in communication systems from Peter the Great Saint Petersburg Polytechnic University, Saint Petersburg, Russia, in 2018. Her current research interests include network energy saving, non-terrestrial communications, and ML/AI-aided network optimization algorithms.

Olga Vikhrova is an Academy Postdoctoral Researcher at Tampere University, Finland. She received her Ph.D. (2021) in Information Engineering from University Mediterranea of Reggio Calabria, Italy, Cand.Sc. (2017) and M.Sc. (2014) in Computer Science from Peoples' Friendship University of Russia. Her current research interests include AI-aided management and optimization of RAN, semantic communications, and energy efficiency in 6G wireless communication systems.

Wei Mao received his B.E. and M.E. degrees from Tsinghua University, Beijing, China, in 2004 and 2007, respectively, and his Ph.D. degree from California Institute of Technology, in 2015, all in electrical engineering. From 2015 to 2017 he was a Postdoctoral Scholar in the department of electrical engineering, University of California, Los Angeles. In 2017 he joined Intel Corporation, Santa Clara, CA, as a research scientist. His research interests include Reinforcement Learning, Communications, Signal processing, Information theory, and Coding.

Shu-ping Yeh is an AI-Applied Principal Engineer in the Wireless System Research Lab at Intel, where she conducts research on wireless broadband technologies. Her current research focus includes open RAN architecture, AI/ML for RAN control and management, network slicing and interworking of multiple radio access technologies within a network. Dr. Yeh has over 10 years of research and development experience in wireless industry and holds more than 30 US patents. She received her M.S. and Ph.D. in Electrical Engineering from Stanford University in 2005 and 2010, respectively.

Omid Semiari (S'14, M'18) received his Ph.D. from Virginia Tech in 2017. He is currently an AI Applied Research Scientist at Intel Labs in Santa Clara, CA. Prior to joining Intel, he was an assistant professor with the Department of Electrical and Computer Engineering at the University of Colorado, Colorado Springs. His research interests include wireless communications (6G), machine learning for wireless networks, distributed learning over wireless networks, and cross-layer network optimization. He is the recipient of several research awards, including the NSF CRII award. He has served as an Associate Editor for the *IEEE Journal on Selected Areas in Communications (JSAC)*, machine learning series. He has also been a TPC member for many IEEE flagship conferences.

Hosein Nikopour is a Principal Engineer, Applied AI Research Scientist and Manager at Intel Labs, with a focus on intelligent and resilient next-generation wireless networks, hierarchical learning (H-learning), graph neural networks (GNN), integrated access and backhaul (IAB) networks, high-frequency bands, and network coding. Prior to his current role, he contributed to wireless solutions for 3GPP cellular standards during his tenure at Huawei and Nortel Canada. Dr. Nikopour is widely recognized for his work on Sparse Code Multiple Access (SCMA) for massive machine-type communications. He holds a portfolio of over 150 patents and has authored numerous publications, which have received more than 6400 citations.

Shilpa Talwar is an Intel Fellow and Director of Wireless Systems Research in the Intel Labs organization at Intel Corporation. She leads a research team focused on advancements in ultra-dense multi-radio network architectures and applications of machine learning and artificial intelligence (ML/AI) techniques to wireless networks. While at Intel, she has contributed to IEEE/3GPP/ORAN standard bodies, including 4G/5G and Open RAN Intelligence Control (RIC) specifications. She is co-editor of book on 5G “Towards 5G: Applications, requirements and candidate technologies.” Prior to Intel, Shilpa held several senior technical positions in wireless industry working on a wide range of projects, including algorithm design for 3G/4G & WLAN chips, satellite communications, GPS, and others. Shilpa graduated from Stanford University in 1996 with a Ph.D. in Applied mathematics and an M.S. in electrical engineering. She is the author of 70+ technical publications and holds 65+ patents.

Sergey Andreev received the Ph.D. degree from TUT in 2012, and the Cand.Sc. and Dr. Habil. degrees from SUAI in 2009 and 2019, respectively. He is currently a Professor of wireless communications and Academy Research Fellow at Tampere University, Tampere, Finland. He is also a Research Specialist with Brno University of Technology, Brno, Czech Republic. He was a Visiting Senior Research Fellow with King’s College London, London, U.K., during 2018–2020, and a Visiting Postdoc with the University of California, Los Angeles, Los Angeles, CA, USA, during 2016–2017. He has (co-)authored more than 300 published research works on intelligent IoT, mobile communications, and heterogeneous networking.