IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 27, 2021, accepted August 4, 2021, date of publication August 12, 2021, date of current version August 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3104322

Access and Radio Resource Management for IAB Networks Using Deep Reinforcement Learning

MALCOLM M. SANDE[1], (Graduate Student Member, IEEE),
MDUDUZI C. HLOPHE[2], (Member, IEEE),
AND BODHASWAR T. MAHARAJ[3], (Senior Member, IEEE)
Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0028, South Africa
Corresponding author: Malcolm M. Sande (u10410903@tuks.co.za)

This work was supported by the Sentech Chair in Broadband Wireless Multimedia Communications (BWMC), University of Pretoria.

ABSTRACT Congestion in dense traffic networks is a prominent obstacle towards realizing the performance requirements of 5G new radio. Since traditional adaptive traffic signal control cannot resolve this type of congestion, realizing context in the network and adapting resource allocation based on real-time parameters is an attractive approach. This article proposes a radio resource management solution for congestion avoidance on the access side of an integrated access and backhaul (IAB) network using deep reinforcement learning (DRL). The objective of this article is to obtain an optimal policy under which the transmission throughput of all UEs is maximized under the dictates of environmental pressures such as traffic load and transmission power. Here, the resource management problem was converted into a constrained problem using Markov decision processes and dynamic power management, where a deep neural network was trained for optimal power allocation. By initializing a power control parameter, θ_t, with zero-mean normal distribution, the DRL algorithm adopts a learning policy that aims to achieve logical allocation of resources by placing more emphasis on congestion control and user satisfaction. The performance of the proposed DRL algorithm was evaluated using two learning schemes, i.e., individual learning and nearest neighbor cooperative learning, and this was compared with the performance of a baseline algorithm. The simulation results indicate that the proposed algorithms give better overall performance when compared to the baseline algorithm. From the simulation results, there is a subtle, but critically important insight that brings into focus the fundamental connection between learning rate and the two proposed algorithms. The nearest neighbor cooperative learning algorithm is suitable for IAB networks because its throughput has a good correlation with the congestion rate.

INDEX TERMS Congestion control, deep reinforcement learning, integrated access and backhaul, millimeter wave, nearest neighbor, resource allocation.

I. INTRODUCTION AND BACKGROUND

The initial deployments of 5G networks and the smart devices that are currently running have emphasized the enhanced mobile broadband and the massive machine-type communications legs of 5G use cases [1]. However, 5G new radio (NR) is continually being developed to provide the foundation for future mobile and wireless networks by supporting other new types of applications. Various technologies that enable the existence of smart mobile terminals in 5G networks, which are expected to access unused spectral bands in an opportunistic manner, have been proposed in literature. These technologies include cognitive radio (CR) terminals, millimeter wave (mmWave) communications, heterogeneous networks, device-to-device communication, energy harvesting, and smart grids [2]. The increasingly densified demand for high throughput and low latency communications is envisaged to bring about a paradigm shift in the design of cellular networks, where smart terminals will perform opportunistic spectrum access through CR technology. From the 5G NR perspective, a smart device should be aware of its environment through being able to sense and identify the various radio frequency activities in its surroundings, and it should be able to learn and make reasonable decisions [3]. It is therefore evident that learning is an essential tool to enable 5G networks to provide better quality of service (QoS) and

The associate editor coordinating the review of this manuscript and approving it for publication was Byung-Seo Kim.

quality of experience (QoE) to end users. Among the requirements to afford users high QoS and QoE is utilizing the available information in a real-time manner based on the network, the devices and the applications, as well as the users in their different contexts [4]. This kind of context awareness introduces the need to integrate multiple parameters in the optimization of network functionalities. This seems to be a design requirement for current and future smart devices, and as such, a number of models that consider context awareness for 5G networks have been proposed [5]–[7]. With the persistent need for integrating multiple parameters, and perform dynamic load balancing, arises the need for optimizing network functionalities. However, the critical aspect of the context awareness problem is the requirement to understand the manner in which mobile and wireless devices perceive, operate, and experience a space.

The increased cell densification of 5G networks makes it difficult and costly for mobile network providers to provide fiber backhaul to every access point in the network [8]. As a result, integrated access and backhaul (IAB) was developed to address this challenge by leveraging the availability of large amounts of spectrum in the mmWave frequencies [9]. In IAB networks, the wireless spectrum is shared between access and backhaul and the sharing is either in-band, i.e. using the same frequency bands, or out-of-band, but on overlapping time slots in order to make optimal use of resources [10]. In a typical IAB network, a slave base station (SBS) sends/receives backhaul data wirelessly through a single direct link or over multiple hops to/from a master base station (MBS). IAB can enable flexible and very dense network deployment without the need for densifying the transport network accordingly, especially when operating in mmWave bands, and it is envisaged to be the most flexible and cost-effective backhaul technology for mobile networks in 5G networks [11]. However, designing an efficient and high-performance IAB network that satisfies 3GPP requirements, such as handling stochastic and bursty data, is still an open research challenge. The stochastic and bursty nature of user data requirements in 5G NR presents itself as a huge challenge to network designers. In addition to the dynamics of the wireless environment, such as random user mobility, fading, shadowing, and path loss effects, the challenge of unbalanced traffic distribution makes it difficult to rely on a single model for solving network optimization problems [4]. To solve the problem of traffic balancing, IAB node coordination is a requisite. In the case of IAB node coordination, efficient signaling exchange among the medium access control (MAC) layers of different IAB nodes is required, considering the rate and latency constraints of wireless backhaul links. For successful IAB node coordination, uplink and downlink interference has to be introduced in case of asynchronous IAB node transmission mode. Thus, adaptive intelligence algorithms that make adaptive decisions on the link and user/base station (BS) scheduling with fairness and half-duplex constraints need to be implemented at the MAC layer of IAB nodes. However, this requires centralized training procedures to be distributed among local IAB nodes. The key requirements for the successful implementation of IAB networks are flexibility and programmability, which involve the incorporation of software defined networking (SDN).

Autonomous resource management and congestion avoidance is desirable in 5G networks in order to continuously satisfy user requirements. By applying SDN, mobile operators can enforce per-user or even per-BS policies, and in case of congestion or link failure, the control plane could quickly reconfigure itself to deal with the available traffic [12]. Resource provisioning schemes and deep reinforcement learning algorithms that are applicable for mixed traffic in virtualized radio access networks have been proposed [13], [14]. The proposed shape-based heuristic algorithms were shown to improve resource utilization and user satisfaction through system-level simulations, which considered varying user requirements. In this work, we aim to improve resource management and user satisfaction through congestion avoidance by applying deep reinforcement learning in mm-wave IAB networks.

A. RESOURCE ALLOCATION IN IAB NETWORKS

Because of the radio resource sharing between access and backhaul, the IAB network architecture requires a different RA approach from other typical wireless standards. This means that end-to-end RA algorithms are most suitable for IAB networks. To this effect, [9] proposed an end-to-end system-level algorithm to improve cell-edge throughput. Through end-to-end simulations, the authors demonstrated the cell edge throughput advantage offered by IAB networks. In their paper, they also highlighted some research challenges that require further investigation. The success of IAB networks is related to cell density, which requires accurate deployment of heterogeneous BSs in order to achieve better network load balancing. BS deployment and network topology are better handled using stochastic geometry and point processes; as a result stochastic geometry-based algorithms were proposed to tackle load-balancing problems. Based on the stochastic geometric framework, [15], [16] proposed dynamic load-balancing schemes for IAB operation in heterogeneous networks. Here, the authors used point process-based models on sub-6GHz networks. The application of stochastic geometry and point processes for dynamic load balancing was extended in the context of mmWave networks in [16] and [17]. The authors in [18] analyzed the performance of three wireless backhaul partition strategies in an IAB network where all the SBSs are wirelessly backhauled over mmWave links. For the same network model, the work of [19] considered the impact of limited backhaul capacity in mmWave IAB networks by analysing the rate coverage probability for two types of resource allocation at the MBS. The authors of [20] have demonstrated the feasibility of in-band wireless backhaul in mmWave bands, and they also presented a baseline scheduling scheme for SBS-SBS communications.

Since CR technology will be integrated into every future technological innovation, every wireless access point must

have a cognitive engine for automated handover. The role of the cognitive engine will be to assist in BS-user equipment (UE) association, whereby UE can instantaneously switch between BSs for better QoS. This is practically impossible, considering that the prospective BS might not immediately have the resources to support the QoS requirements of the newly arriving users. The resulting handover delays may lead to transmission delays, hence deteriorating the QoE. Many contributions in user association begin by tacking the problem of aggregated interference generated by users [21]. Such problems are usually modeled as multi-agent systems based on the IEEE 802.22 standard for wireless regional area networks. In all the existing contributions, the BSs are generally the agents in control of admitting users, and the problems are usually modeled as multi-agent systems. Real-time multi-agent reinforcement learning techniques known as decentralized Q-learning are used to manage the aggregated interference generated by multiple users. To this effect, two scenarios are considered to enable the multi-agent systems to learn, i.e., (i) agents having complete or partial information about the environment, and (ii) agents directly interacting with the surrounding environment in a distributed fashion. As a result, the resulting spectrum management framework improves the spectrum utilization efficiency while increasing the energy efficiency, as reported in [22]. However, since balancing the spectral efficiency and energy efficiency has become a critical challenge in current heterogeneous and resource-constrained networks, channel characteristics and energy efficiency are analyzed using joint channel selection and power control spectrum decision algorithms based on distributed Q-learning. In this way, the selection of the learning strategy designed to solve the optimization problem introduces distributed strategy estimation. This was the case in [23], where the authors formulated a channel access problem using a non-cooperative game. In their contribution, each channel can only be used by one user at a time. However, the channel switching distance was only limited to a certain scope by considering transmission delays, and the optimal access policy was dependent on the long-term behavior of other users in the network. The solution to this non-cooperative game was achieved using a multi-agent Q-learning algorithm, which required neither prior knowledge of channel dynamics nor negotiations among players.

As opposed to traditional wireless network environments, where BSs employ static spectrum allocation strategies such as full-reuse or fixed orthogonal allocation methods in order to ease the system computation and implementation complexities, in IAB networks such spectrum re-use schemes are inefficient. For example, in ultra-dense IAB environments, where there may be severe co-tier and cross-tier interference among neighboring BSs, static spectrum allocation schemes are inefficient. In such ultra-dense environments, the rate of UEs associated with an SBS is determined by how the total wireless bandwidth is split between the backhaul link and the access link. This makes the achievable throughput sensitive to the spectrum allocation strategies applied such that when the number of devices per BS increases and more spectrum resources become available, the solution space for spectrum allocation increases exponentially. In the reinforcement learning (RL) community, it is believed that the best way to address this problem is to employ model-free RL algorithms. As a result, the authors in [24] used a scalable and model-free RL algorithm to handle the large state spaces. This was done to provide good approximation of the Q-values and perform dynamic spectrum allocation in order to maximize the sum log-rate, while satisfying the UE demands. In order to solve the optimization problem, two deep reinforcement learning (DRL) algorithms were applied, i.e., double deep Q-learning networks and actor-critic spectrum allocation.

Considering the aforementioned intelligent developments in IAB schemes, it is believed that the IAB modeling problem requires artificial intelligence (AI) strategies. Even though the application of AI strategies has not been intensively explored in IAB networks, recurrent neural networks (RNNs) are one of the state-of-the-art models that are favorable owing to their ability to store information over extended time intervals. Using RNNs, historical information can potentially be used to predict traffic from all user groups and to facilitate the optimization of future transmission time interval configurations. The disadvantage of the RNN is that even if one knew all the relevant statistics, tackling the RA problem in IAB networks in an exact manner would result in a partially observable Markov decision process with large state and action spaces. In this way, the complexity of the problem is compounded by the lack of prior knowledge regarding the stochasticity of traffic as well as the unobservable channel statistics at each BS node, which makes it generally intractable.

B. RESEARCH MOTIVATION

From the reviewed research contributions, it transpired that the dynamic RA problem and data transmission resource configuration still remains a less investigated problem in IAB networks. Existing algorithms focus on finding low-cost routes for traffic to reach the MBS, without finding ways to minimize resource exhaustion at the current BS. A framework that avoids resource exhaustion must have the capability to adapt to different learning mechanisms as well as real-time system requirements effectively. Future mobile and wireless networks' operational spaces will be very diverse and will vary significantly, which will lead to scenarios not postulated during the design phase. Because of the unpredictability of future wireless environments, rule-based decision-making that selects decisions directly from training may not be ideal. As a result, it may not be effective to design a priori cost functions and then solve optimal control problems in real-time. This would be detrimental. For this purpose, the decision maker of an IAB system has to be implemented using a deep neural network DNN) in order to provide action choices for any given state of the system. DNNs are crucial for acting optimally in highly stochastic and dynamic environments such as IAB networks, where the value of taking an action

depends on future actions and states. Therefore, it is at this point where DRL strategies become an attractive alternative.

C. RESEARCH CONTRIBUTIONS

The objective of this article is to obtain an optimal policy under which the transmission throughput of all UEs is maximized under the dictates of environmental pressures such as traffic load and transmission power. Using a DRL strategy, a logical combination of wireless channel gains and traffic load is used to influence the system to provide better QoS without upsetting power consumption. The main contributions of this article are summarized as follows:

- **Adaptive Congestion Control:** A heterogeneous broadband access network is proposed with investigations based on IAB network transmission. The effects of congestion on the link layer behavior are handled by defining the levels of congestion as the exact values of system utilization. SBS traffic load measurement is implemented using an M/G/1 queuing model with an ergodic arrival process. The optimization problem was formulated as a QoS maximization problem that places distinct importance on estimating service-level pressure on the SBS. This means that the system must be able to learn context-related behaviors and execute appropriate actions within a reasonable time by efficiently leveraging the feedback from the output.
- **DRL Adaptive Learning Scheme:** Because of the variation in UE service rates and their generation rates, a DRL algorithm that uses an online RL approach is proposed. Here, the system continuously monitors the congestion rate of the SBS, where the output of the DNN agent is a vector of possible transmission powers. Then, based on the already allocated resources, which define the current traffic load, the decision function of the DRL generates optimal actions using a policy π_θ. At this point, the need for a perfect control system is pressingly important, such that the computation of the DRL agent then focuses on avoiding a situation whereby the system is placed under immense pressure to increase the packet departure rate by raising the transmission power, which may result in high energy consumption. Under this approach, the congestion rate, throughput, and the quality of experience are evaluated. In order to realize a reliable system, the proposed approach is evaluated using two algorithms, i.e., (i) individual learning algorithm, and ii) nearest neighbor cooperative algorithm.

The rest of the paper is organized as follows: Section II describes the proposed IAB network model and gives detailed discussions of the queuing model and state- and action-space definitions. Section III discusses the mathematical formulation of the problem and presents the optimization problem description. Section IV describes the proposed adaptive learning scheme and provides a detailed solution using a DRL strategy. In Section V, the performance of the proposed adaptive learning scheme is evaluated using the two proposed algorithms and simulation results are presented and discussed. The concluding remarks of the performance analysis of the proposed learning scheme are given in Section VI.

II. PROPOSED SYSTEM MODEL

Considering the 3GPP heterogeneous broadband access network access for multi-hop IAB networks [25], the uplink transmission of a two-tier IAB network is investigated. Here, the MBS, indexed by m_0, is equipped with an omni-directional antenna. A set $\mathcal{M}^- = \{1, 2, \ldots, M\}$ of IAB nodes is uniformly deployed within its coverage area and connected to it via mmWave backhaul. Therefore, let $\mathcal{M} = m_0 \cup \mathcal{M}^-$ denote the set of all BSs such that $|\mathcal{M}| = 1 + M$. It is assumed that each IAB node has a representative group of UEs associated with it such that $\mathcal{K} = \{1, 2, \ldots, K\}$ denotes the set of UEs associated with each IAB node according to a call admission scheme [26], as shown in Fig. 1. Assuming the non-stand alone deployment scenario of the 5G NR [27], the convenience of using physical resource blocks as a measure of physical radio resources is used. In line with the IAB setup, the nodes are assumed to be full-duplex capable, all SBS-SBS links are symmetrical, and proper bandwidth partitioning according to access and backhaul is adhered to. This means that each IAB node is assumed to be equipped with two antennas, i.e., one for access to serve its associated UEs, and another for wireless backhaul with the SBS. The effects of SBS congestion on the link layer behaviour are used to evaluate achievable throughput and user satisfaction in terms of system utilization. Fig. 1 also shows the distribution of environmental states to be input into the algorithm, i.e., the average waiting time, and average SBS load. These are fed as inputs into the DNN agent, and based on the computation of the DNN agent, optimal transmit power or change of learning scheme is the output/action. The environmental reward, which results from the optimal action, leads onto another environmental state, and so on.

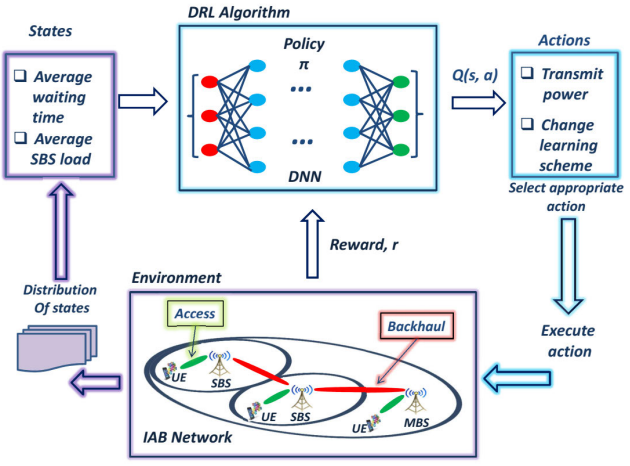

FIGURE 1. Illustration of a typical multi-hop IAB network that implements a DRL algorithm.

A. NOTATIONS AND DEFINITIONS

In order to enhance readability and ease of exposition, the notations used in this article and their descriptions are listed in TABLE 1 below.

TABLE 1. List of notations and their definitions.

Notation	Description
$\mathbb{E}[\cdot]$	Mathematical expectation operator of $[\cdot]$
$\overline{x}$	Mean value of the time-series x
$M \in \mathcal{M}$	Total number of SBSs in the set of all SBSs, $\mathcal{M}$
$K \in \mathcal{K}$	Total number of UEs in the set of all UEs, $\mathcal{K}$
λ	UE arrival rate
p	The selected transmission power
z	The required transmission throughput
T	The length of a finite horizon
B; B_a	Total system bandwidth; The access bandwidth
B_b	The backhaul fraction of total system bandwidth
η	The bandwidth partition factor
L; T_s	Packet length; The symbol duration
h_{km}	Channel gain between k^{th} UE and m^{th} SBS
γ	The signal-to-interference-plus-noise ratio
γ^*	The target signal-to-interference-plus noise ratio
C	Transmission buffer capacity
ρ^*; ψ; $\Delta\rho$	The SBS load status; The load threshold; Load trend
$r_{k,m}$	The achievable transmission rate between UE k and SBS m
ω	Decay factor of load trend
ζ	The power allocation condition
α	Learning rate
β	Discount factor

B. QUEUING MODEL AND TRAFFIC LOAD MEASUREMENTS

The average SBS load is a result of traffic injected by K sources, i.e., UEs, at a rate, $\lambda(t)$, and arriving at the SBS in a collective arrival process, $\Lambda(t)$, denoted as follows:

$$\Lambda(t) = \sum_{t=t_0}^{T-1} \lambda(t), \quad t_0 = 0, 1, \ldots, T-1 \tag{1}$$

which represents the arrival flow, whose bivariate extension for the range $t_0 \leq t \leq T$ is defined as $\Lambda(t_0, t) = \Lambda(t) - \Lambda(t_0)$, up to a time horizon T. Assuming that the system is monitored at discrete time intervals, $t = 0, 1, 2, \ldots$, the duration $\Delta t = t_1 - t_0$ represents a single time slot. To guarantee the existence of stationary limits, a discrete-time $M/G/1$ queuing system with a stationary and ergodic arrival process is assumed [28]. Assuming that the $M/G/1$ defines the arrival and service process of a transmission buffer capable of handling C packets, based on the stationary and ergodic assumption, the system utilization factor can be defined as follows:

$$u(t) = \frac{\mathbb{E}[\lambda(t)]}{C}, \tag{2}$$

where $\mathbb{E}[\lambda(t)]$ is the long-term expectation of the packet arrival process. The consumed capacity is the monitored condition, whose trend is periodically tracked using an exponentially weighted moving average defined as follows [29]:

$$\overline{\Delta\rho} = \omega \cdot \Delta\rho(t) + (1-\omega)(\rho(t) - \rho(t-1)), \tag{3}$$

where $\overline{\Delta\rho}$ is the mean value of the load behavior, with $0 < \omega < 1$ representing the decay factor, which is selected through adaptive weights. The occupancy level marker between two successive time slots is given by $(\rho(t) - \rho(t-1))$ [30].

C. DEFINING STATE AND ACTION SPACES

In response to the variation in request generation rate from the UEs, the change in the distribution of their requests, and the SBS processing rate, the state space of the system can be defined in accordance with the average waiting time and the average SBS load. The implementation of SBS traffic load measurements requires that the state and action vectors, as well as the reward, be defined. As stated above, the computation of control variables can only be performed at the beginning of the first time slot; $t = 0$ is regarded as the time when the start-state is defined, i.e., s_0. With the service rate and transmission delays assumed to be independent and generally distributed with respect to the service rate, the state space can be defined as follows:

$$\mathbf{s} = \{C(t), \rho^*(t)\}, \tag{4}$$

where $C(t)$ is the time-average number of packets in the system up to time t, which tends to the steady state as $t \to \infty$. Since the state space reveals how much pressure is endured by the SBS's transmission buffer, the computation of the DNN agent determines either the transmission power or change of learning mechanism as the output. Therefore, the action space can be represented in vector form as follows:

$$\mathbf{a} \triangleq < p(t), z(t) >, \tag{5}$$

where $p(t)$ is the selected transmission power and $z(t)$ is the required transmission throughput. In order to arrive at the required throughput, the selection of the optimal transmission power takes precedence.

III. MATHEMATICAL PROBLEM FORMULATION

Let $k \in \mathcal{K}$ represent the generic progression in progress connection so that the range of admitted UEs can be represented as $[1, K]$. Assuming uniformity of channel fading within one sub-channel and a difference on the other sub-channels, the signal-to-interference plus noise ratio (SINR) between the k^{th} UE and the m^{th} SBS can be expressed as follows:

$$\gamma_{k,m} = \frac{p_k g_{k,m}}{\sum_{j \in \mathcal{M}^-} p_k g_{kj} + N_0}, \quad \forall k, m \tag{6}$$

where p_k is the transmission power, $g_{k,m} = \alpha_{k,m} h_{k,m}$; with α_k being the distance-dependent fading coefficient and $h_{k,m} = \exp(1)$ being the frequency-dependent small-scale fading [24]. The first term in the denominator represents the co-tier interference, and the term N_0 represents the white Gaussian noise spectral density. Assuming that self-interference cancellation abilities exist in both UEs and SBSs, the achievable instantaneous transmission rate between UE k and SBS m is determined as follows:

$$r(\gamma_{k,m}) = B_a \log_2 \left(1 + \gamma_{k,m}\right), \tag{7}$$

where the term B_a denotes the access bandwidth. For all the admitted flows, the SBS has the problem of finding the optimal transmission power that corresponds to the bitrates of the corresponding QoE requirements. In this case, the buffer occupancy and power allocation are inextricably intertwined since proper power control is required to increase the service rate and avoid congestion, while saving system energy. Therefore, depending on the current traffic load, the load at the SBS can be defined as follows [31]:

$$\rho^*(t) = \frac{C_i(t)}{\sum_{k=1}^{K} B^* \log_2(1+\gamma_{k,m})}, \tag{8}$$

where $C_i(t) \leq C$ is the capacity required by the admitted UEs, and B^* is the sub-channel width. The average service time of all UE requests has to be minimized, which means that admission and processing delays should be at their minimum. Thus, the average service time minimization problem becomes a QoS maximization problem, and it is distinctly important to have a perfect estimate of the service pressure at the SBS. Therefore, the sum rate of the admitted flows can be defined as

$$Q_k(r(\gamma^*, [0,T])) = \sum_{t=1}^{T} r(\gamma(t)), \quad \forall k \in \mathcal{K} \tag{9}$$

where $[0, T]$ is the overall time interval during which the system performance is being monitored. For purposes of simplification and ease of computability, it is assumed that all the admitted flows have the same weighting at the SBS. Then, the objective is to obtain the optimal policy that maximizes the transmission throughput of the UEs connected to the SBS, defined as follows:

$$\mathbf{P} = \arg\max_{\gamma^*} Q_k(r(\gamma^*, [0,T])), \quad \forall k \in \mathcal{K} \tag{10}$$

$$\begin{aligned} \text{subject to } \mathbf{C1}: & \sum_{k=1}^{K} r_{k,m}^{req} \leq r(\gamma(t)), \\ \mathbf{C2}: & \, r(\gamma(t)) \geq \Omega(\gamma^*, [0,T]), \quad \forall k \in \mathcal{K} \\ \mathbf{C3}: & \, \rho^*(t) < \psi \cdot B_a, \quad \forall k \in \mathcal{K} \\ \mathbf{C4}: & \, \zeta_k(\gamma^*, [0,T]) \leq 1-\epsilon, \quad \forall k \in \mathcal{K} \end{aligned} \tag{11}$$

where $\mathbf{P}$ represents a function $f(x) = Q_k(r(\gamma^*, [0,T]))$ that is designed to obtain maximum network utility, where γ^* represents the target SINR for achieving proportional transmission fairness, $r(\gamma^*, [0,T])$. According to proportional fairness, $r(\gamma^*, [0,T]) = log\gamma^*$, which is supported by constraint **C2**, indicating that the QoS requirements (7) of the admitted UEs must be met. Based on the M/G/1 model, the satisfaction rate, $\Omega(\gamma^*, [0,T]) \approx \sigma/T_s$ is measured on a scale of 1 to 5 and is assumed to be general, with T_s being the symbol duration. The constraint **C1** ensures that the bandwidth requirement for all UEs is kept within the allocated access bandwidth B_a, with $r_{k,m}^{req}$ being the required data rate. The constraint **C3** ensures that the traffic load, ρ, does not exceed the threshold of the maximum capacity to avoid congestion and a subsequent decline in the QoS [29]. This condition avoids the classical congestion collapse by tracking the congestion behavior. Here, an SBS is allowed to admit UEs as long as it can still cope with the requirements, until it hits the threshold where the it becomes overloaded and goodput starts to decrease. Lastly, the constraint **C4** ensures that every action exploration is kept within acceptable powers, where $\zeta_i(\gamma^*, [0,T]) = \frac{\gamma}{\gamma+1}$ is the power allocation condition, and ϵ is the exploration parameter. When congestion has reached the threshold in **C3**, the exploration of transmission powers is instantiated within the power budget, without upsetting the energy consumption of the system.

IV. PROPOSED DEEP REINFORCEMENT LEARNING SOLUTION

In designing a better solution to solve the optimization problem in (10), a finite-source traffic model is assumed. The finite-source traffic model is based on both the thinning process and fading conditions and is assumed to be appropriate for newly originated traffic [32]. Therefore, the input sample that is clamped to the input of the DNN is computed based on $[5 : 5 : K]$ sources injecting packets into the SBS transmission buffer. Considering only the access part of the IAB network, the input, output, and feedback processes of the DRL algorithm define the appropriate learning scheme to be used. Under each learning scheme, the network status is relatively fixed under the assumption that the application environment is known. In this way, the distribution of states, actions, and rewards define the dynamics of the IAB network and node behavior can be estimated using the pressure on the communication and computation resources. Thus, the proposed DRL algorithm consists of a hierarchical structure that computes the solution for $Q(\gamma^*, a) \approx \sigma/T_s$. For the sake of convenience, it is assumed that the state and observation overlap perfectly so that it is easier to apply the DRL strategy to this problem.

For any admitted UE, there has to be adequate computational resources allocated to it such that the required QoS is met. Thus, using adaptive modulation and coding, the peak data rate, and the gains from other transmitting technologies within the cell, the spectral efficiency used in [33] is adopted. Then, using the relationship between the state and action spaces used in [34], the reward function can be formulated as follows:

$$R(t) \triangleq \sigma = \lceil z(t) L T_s / \Delta t \rceil, \tag{12}$$

where $\lceil \cdot \rceil$ means that the transmission throughput has to be calculated using the minimum power required to transmit $\lceil x \rceil$ bits per second, $z(t) \in Z$ represents the transmission throughput in packets/time slot and L represents the packet length. Given that the packet throughput determines the number of bits per symbol, using the channel state, the system can choose the transmission power from which the reward in (12) can be determined. Here, the DNN agent generates an optimal action $a^* \in \mathcal{A}$, which is the power allocation action that is used to obtain the solution to (10). The action-value function

stored in the Q-table at time t is the one that will be used to select an action according to the current state through policy π^*, as follows:

$$\pi^*(a|s) = \left\{ \begin{array}{ll} 1, & \text{if } a = \arg\max_{a\in\mathcal{A}} Q^{\pi}(s,a) \\ 0, & \text{otherwise,} \end{array} \right\}. \quad (13)$$

After approximating the Q-function, a common off-line algorithm that takes a greedy search is used to find the optimal stochastic policy function $\pi^*(a|s)$ [36]. A random variable, $Q(s_t, a_t)$, is then considered for the estimation of $Q(\gamma^*, a)$ as follows:

$$Q(s_t, a_t) = R_t + \beta^t \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}), \quad (14)$$

where R_t is the reward computed according to (12), $\beta^t \in [0, 1]$ is the discount factor and s_{t+1} and a_{t+1} represent the next state and action, respectively. Thus, at each coherence time, the agent builds its state using the information collected from its transmission queue, and with this information restricted to the QoS of each UE, the system computes the rate level of each UE according to (7) and (10). To this point, a sequence of transition probabilities is created to describe the transitions from the current state s_t to the next state s_{t+1}, current actions a_t to next actions, current rewards to future rewards as follows:

$$P(s_t|a_t) = p_p(p_{tx}|[g_{k,m}, x_t], h_{k,m}) p_g(g_{t-1}|g_t), \quad (15)$$

where x_t is the current power management. The system now has a Markov property, such that the representation function is a Markov model of the IAB environment with the tuple, $(s_t, a_t, r_t, p_t, s_{t+1}), t \in T$. Here, the current channel state, $h_{k,m}$, taken from the current measurement of the received signals is expressed using the channel transition distribution, $p_g(g_{t-1}|g_t)$. Then, the reward function $R_t(s_t, a_t) \in \mathcal{R}$ is represented as follows:

$$R_t(s_t, a_t) = \theta \cdot Q(s_t, a_t), \quad (16)$$

where, the term θ represents the interference power constraint, which is an essential factor in both power management and in the estimation of $Q(\gamma^*, a)$. It should be noted that θ is controlled by the $p_p(p_{tx}|[g_{k,m}, x_t])$ in (15) above. Using (14), and assuming that the state-action value function of the true state is $Q(s_t, a_t)$, and is independent of the reward features of s_t and a_t, (16) can be reformulated as follows:

$$R_t(\gamma^*, \hat{p}) = \theta \cdot Q(\gamma^*, \hat{p}), \quad (17)$$

such that the solution to (10) can be summed into a state-value function over a finite horizon as follows:

$$Q_t^{\pi}(\gamma^*, \hat{p}) = \mathbb{E}\left[\sum_{t=1}^{T} \beta^t R_t(\gamma^*, \hat{p})\right], \quad (18)$$

which is the expected value of the reward based on the transmission power, π is the optimal policy for all states and actions defined by a value function, which is computed to obtain the Bellman optimality equation [36]. The expression $\beta^t R_t(\gamma^*, \hat{p})$ is the discounted reward at time step t, the sum is over a finite horizon, T, and $\beta^t < 1$ because the agent is interested long-term returns. Thus, $\sum_{t=1}^{T} \beta^t R_t(\gamma^*, \hat{p})$ is the discounted reward obtained over T. Since the optimal values of $R_t(\gamma^*, \hat{p})$ depend on the physical conditions of the environment as well as the policy followed by the RL strategy, the objective of maximizing (18) can be represented as follows:

$$Q_t^{\pi}(\gamma^*, \hat{p}) = \max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} \beta^t R_t(\gamma^*, \hat{p})\right]. \quad (19)$$

Using the intuitive definition of the Bellmann optimality equation, which expresses the fact that the value of a state under an optimal policy must be equal to the expected return for the best action taken from that state, Q^* can be expressed as follows:

$$\begin{aligned} Q^*(s,a) &= \mathbb{E}\left[R_{t+1} + \beta \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1})\right] \\ &= \sum_{s_{t+1}, R_t} p(s_{t+1}, R_t|s_t, a_t)[R_t \\ &\quad + \beta \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}]. \end{aligned} \quad (20)$$

Here, the expression $\sum_{s_{t+1}, R_t} p(s_{t+1}, R_t|s_t, a_t)$ represents the transition probability. In the Bellman optimality equation, the value of a state can be decomposed into the immediate reward and the discounted value of the successor state, s_{t+1} [37]. The Bellman optimality equivalent to Q^* can be derived by assuming perfect knowledge of (20). By assuming perfect knowledge of Q^*, and that the deterministic policy π is optimal, (13) holds. From the state-value function in (18), the Bellman optimality equation for Q^{π} is computed using the value function as follows:

$$\begin{aligned} &Q^{\pi}(s,a) \\ &= \mathbb{E}_{\pi}\left[\sum_{t=1}^{\infty} \beta^t R_{t+1}|s_t = s\right] \\ &= \mathbb{E}_{\pi}\left[R_{t+1} + \beta \sum_{t=1}^{\infty} \beta^t R_{t+2}|s_t = s\right] \\ &= \sum_{a} \pi(a|s) \sum_{s_{t+1}} \sum_{R_t} p(s_{t+1}, R_t|s_t, a_t) \\ &\quad \left[R_t + \beta \mathbb{E}_{\pi}\left[\sum_{t=1}^{\infty} \beta^t R_{t+2}|s_{t+1} = s_{t+1}\right]\right] \\ &= \sum_{a} \pi(a|s) \sum_{s_{t+1}, R_t} p(s_{t+1}, R_t|s_t, a_t)\left[R_t + \beta v_{\pi}(s_{t+1})\right]. \end{aligned} \quad (21)$$

Using the relationship $V^*(s) = \max_{a\in\mathcal{A}} Q^{\pi *}(s_t, a_t)$ the objective to maximize (18) can be computed as follows:

$$\begin{aligned} V^*(s) &= \max_{a\in\mathcal{A}} Q^{\pi *}(s_t, a_t) \\ &= \max_{a} \mathbb{E}_{\pi^*}\left[\sum_{t=0}^{\infty} \beta^t R_{t+1}|s_t = s, a_t = a\right] \end{aligned}$$

$$= \max_{a} \mathbb{E}_{\pi^*}\left[R_{t+1} + \beta \sum_{t=0}^{\infty} \beta^t R_{t+2} | s_t, a_t\right]$$
$$= \max_{a} \mathbb{E}_{\pi^*}\left[R_{t+1} + \beta V^*(s_{t+1}) | s_t, a_t\right]$$
$$= \max_{a \in \mathcal{A}} \sum_{s_{t+1}, R_t} p(s_{t+1}, R_t | s_t, a_t)\left[R_t + \beta V^*(s)\right], \quad (22)$$

which is the Bellman optimality equation for V^*.

For small problems such as spectrum sensing, one may start by making arbitrary assumptions for all Q-values and then update them through trial-and-error as the policy progresses towards convergence. In this case, the update and choice of action is done randomly, and as a result, the optimal policy may not represent a global optimum [38]. However, in mobile and wireless communications, the problems can become large-scale, with discrete states and actions. Discrete state transition probabilities need to be defined explicitly based on a state space that is large. Constructing and storing a set of Q-tables for large problems in a dynamic operating environment becomes a daunting computational task since the number of possible states grows exponentially with the number of future states and actions to be calculated. As a result, the amount of memory that is required to save and update the Q-tables increase as the number of possible states increase, and the amount of time required to explore each state to create the required Q-table becomes unrealistic. Due to the challenge of scaling and complexity in the traditional RL strategy, advanced strategies such as DRL help to tackle this challenge [39]. Thus, in order to proceed with the RA problem in IAB networks, the RL strategy is combined with the deep learning technique. Thus, we propose a DRL strategy that is applicable in IAB networks, which applies the successes of DNNs.

At this point, let a conditional prior probability vector $\mathbf{P}$ be defined as a sufficient statistic substitute for the state, s_t, such that (19) can be reformulated as follows:

$$Q_t^{\pi}(\mathbf{P}, a_t) = \max_{\pi} \mathbb{E}\left[\sum_{t=1}^{T} \beta^t R_t(P_{ij}, a_t)\right], \quad (23)$$

where the term P_{ij} represents the inter-state transition probability. Then, using dynamic programming, this problem can be solved by finding the state-value function, $Q_t^{\pi}(\mathbf{P})$, as follows:

$$Q_t^{\pi}(\mathbf{P}) = \max_{a_t}(R_t(P_t, a_t) + \beta \mathbb{E}\{Q_{t+1}(P_{t+1}|P_t)\}), \quad (24)$$

where P_t and P_{t+1} represent successive transition probabilities. Then, using (22), the Bellman optimality equation for different channel variations can be obtained explicitly. Assuming that a limit on T exists, the optimal transmission rate can be reformulated as follows:

$$\bar{r}_t(\gamma^*, \cdot) = \lim_{t \to \infty} \sum_{t=1}^{T} r_t(\gamma^*, \cdot). \quad (25)$$

Then, assuming that the interference caused by other technologies is stationary and ergodic, and P is a function of I, this limit exists with a probability of one and

$$\mathbb{E}[\bar{r}] = \mathbb{E}[r(I), P(I)]. \quad (26)$$

Using the Labesgue-Stieljies integral [40], (26) can be converted to a constrained problem as follows:

$$\int_0^{\infty} r(\gamma^*, P(\gamma^*)) \cdot dP(I \leq \gamma^*) \geq r, \quad (27)$$

where the function P solves the problem into a relatively unconstrained power management function as follows:

$$\min_{P \geq 0} \int_0^{\infty} P(\gamma^*) \cdot dP(I \leq \gamma^*). \quad (28)$$

However, one should note the underlying assumption that the signal processing at the SBS is sufficient to provide an accurate estimate of the interference power, using the total received power.

A. TRAINING OF THE DNN AGENT

It is assumed that at each transmission time interval, the SBS is required to offer a transmission throughput of z bits per symbol in order to meet the QoS requirements of K admitted flows. Using the system state, $s_t \in \mathcal{S}$, the DNN agent has to determine the minimum power required to transmit $\lceil z \rceil$ bits per symbol. Given that the DNN consists of $N : i = 0, 1, 2, \ldots, n$ layers, $i = 0$ refers to the input layer, and $i = n + 1$ to the output layer. The input and output layers are separated by four hidden layers, and the training process of the DNN consists of the two stages described below. Firstly, the DNN's parameter θ is initialized with zero-mean normal distribution. The DNN agent then takes the inputs as a sample vector $\mathbf{x}$ into the first hidden layer. The feed-forward network goes through a procession of hidden layers with hidden features defined by $h = f(x, \theta)$. Here, the value of the j^{th} computational unit of the i^{th} layer of the DNN architecture is denoted as $h_j^i(a)$. In each link between two successive computational units of different hidden layers, is an assignment of a weight W_{jk}, while on the computational node itself a default activation internal bias b_j^i is assigned. With the weight and bias assignments, a rectified linear unit (ReLU) activation, $ReLU(x) = \max(0, x)$, gives the node the necessary "firing" ability to compute the loss function using the input sample from the previous layer. The ReLU is chosen to be used as the activation function for the hidden layers because of its delinquency in solving the vanishing gradient problem, owing to its ability to transmit the error better than the prevalent sigmoid function. This activation function is good for power control DNN applications, especially when employed in the hidden layers because the power values are greater than zero. This computational pattern continues through the different layers of the hidden section of the DNN until the output layer.

With the role of the output layer being to provide additional transformation to the hidden layer outputs, in order to complete the task of the DNN, the design constraints need to be satisfied. Since the output of the last hidden layer has to be thresholded in order to obtain a valid probability at the

output, this requires a careful effort. With y being the output, the probability of the last hidden layer is denoted as follows:

$$P(y=1|x) = \sigma_{relu}(a_j) = \max\{0, \min\{1, a_j^i(\mathbf{a})\}\}, \quad (29)$$

where $a_j^i(\mathbf{a}) = \sum_k W_{jk}^i \hat{h}_k^{i-1} + b_j^i$, so that when $W_{jk}^i \hat{h}_k^{i-1} + b_j^i$ strays outside the unit interval, the gradient of the output of the DNN would not be zero with respect to its parameters. At the output layer, a logistic activation function is applied as follows [31]:

$$\hat{h}_j^i(\mathbf{a}) = \Phi_{sig}(a_j^i), \quad \text{where} \quad \Phi_{sig}(a) = (1+e^{-a})^{-1} \quad (30)$$

is the sigmoidal function. During each step of the forward procession just described, the weights and biases are updated after each forward computation by propagating the weights to the previous layer, before the forward procession continues to the next layer. This is done by computing the weight updates using the delta rule in order to adjust the weights of the DNN, so as to minimize the loss value, which depends on the type of loss function that is employed. Under the current policy, π_{θ_t}, the DNN outputs a relaxed action $\hat{\mathbf{x}}_t$, which can be represented by a parameterized function

$$\hat{\mathbf{x}}_t = \Phi_{sig}(\mathbf{h}_t), \quad \text{where } \hat{\mathbf{x}}_t = \hat{x}_{t,i}, \quad (31)$$

which represents the candidate action $\mathbf{x}_k$ representing the i^{th} entry of $\hat{x}_t$, such that the power allocation action at the output layer has to satisfy $\hat{\mathbf{x}}_t \in (0,1)$. Therefore, using a stochastic gradient descent algorithm, the resulting output is computed as follows:

$$\mathbf{o}(\mathbf{x}) = \hat{\mathbf{h}}^{n+1}(\mathbf{x}) = \Phi_{sig}(a^{n+1}(\mathbf{x})), \quad (32)$$

which is the required power allocation that the DRL algorithm will use to improve the QoS and also support context awareness in the IAB network.

B. THE LEARNING POLICY

At the output of the DNN, the DRL strategy adopts the same learning policy, π, parameterized by θ, i.e., π_{θ_t}, to guide it in learning the best power allocation solution, as follows:

$$\pi_{\theta_t} : p_j \rightarrow a^*, \quad (33)$$

which represents the agents' behavior by directing it on how to choose actions, i.e., optimal power, to guide the system to the best solution of (10). Here, an ϵ-greedy exploration approach is used to learn the best action among the candidate actions output by the DNN and it is assigned as shown in (33). The greedy action selection defined in (13) is repeated here for convenience

$$\mathbf{a}_t^* = \arg\max_{\mathbf{a}_i \in \mathbf{a}_k} Q^*(\mathbf{h}_t, \mathbf{a}_i), \quad (34)$$

where the learned action $\mathbf{a}_t^*$ has the highest value in the Q-table used to achieve reward, and transitions to the next state s_{t+1}. After computing $y = Q(s,a)$ of executing action a under the state s, the packet throughput, which is dependent on the transmission power, is computed via the Q-learning method that was proven to converge effectively to an optimal solution for this problem in [30]:

$$Q^*(\gamma^*, \cdot) = R_t(\gamma^*, \cdot) + \beta^t \sum_{s \in \mathcal{S}} P(s_{t+1}|s_t, a_t) \max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}), \quad (35)$$

where $P(: \mid :, :)$ is the probability measure governing state transitions, with state-action pairs that update the Q-table at each time step to approach an optimal Q-value. The agent then transits to the next state s_{t+1} and updates the corresponding new Q-value as follows:

$$Q(\gamma^*, \cdot) \leftarrow \underbrace{Q(\gamma^*, \cdot)}_{\text{old value}} + \underbrace{\alpha_t(R_t(\gamma^*, \cdot) + \beta^t \max_{a_{t+1}} \underbrace{Q(s_{t+1}, a_{t+1}) - Q(\gamma^*, \cdot)}_{\text{temporal difference}})}_{\text{learned value}}, \quad (36)$$

where $\alpha_t \in [0,1]$ is the learning rate parameter, which controls the convergence rate of the algorithm. The reward, $R_t(\gamma^*, \cdot)$, is then awarded to the best action leading to an optimal, $Q(\gamma^*, \cdot)$. The difference in the Q-value of the learned action in (36) from the next Q-value, $Q(s_{t+1}, a_{t+1})$, is updated every time slot to give the temporal difference (TD) [36]. As a result, the TD becomes the updating rule, such that the learned value at the next time step, i.e., the immediate reward, is defined as follows:

$$R(\gamma^*, \cdot) + \beta^t \max_{a'} Q(s', a'). \quad (37)$$

In this analysis, the system learns to provide the best association and to deliver guaranteed QoS to the associated UEs. The system relies only on the status of the transmissions and its buffer status to learn the appropriate action values that reliably deliver successful transmissions (see constraint **C3**). The objective of the proposed algorithm is to operate at equilibrium and to grant each flow at least the capacity necessary to meet its QoS requirements without upsetting the cost of transmission. The proposed algorithm for evaluating the radio resource management using individual learning is outlined in **Algorithm 1**.

When an SBS not able to correctly observe and learn its environment, information exchange with neighbors help improve their learning processes. In this context, the neighboring SBSs teach and learn from each others' experiences through comprehensive consultation. Locating the nearest SBS according to the Euclidean distance is because the nearest neighbor algorithm critically depends on metric spaces. The proposed nearest neighbor cooperative algorithm is outlined in **Algorithm 2** below.

C. ALGORITHMIC ANALYSIS AND COMPUTATIONAL COMPLEXITY OF REACHING THE REWARD

In the case of the proposed DRL strategy, the DNN helps with the approximation of the expected discounted sum of future rewards for a given state-action pair. Here, a fixed

Algorithm 1: Proposed Individual Learning DRL Algorithm

Input: Bandwidth, B; Exploration policy, π; Learning rate, α_t; Discount factor, β^t
Output: Reward, $R(\gamma^*, \cdot)$
01: Initialize θ_t with zero-mean normal distribution
02: Populate the Poisson arrival distribution
03: Set the environmental state s_t to s_0
04: **For** $t = 1 : T$ **do**
05: Input sample vector of channel gains to DNN
06: Train DNN model to obtain output vector **y**
07: Assign $a \leftarrow y$
08: Take a greedy action according to (13)
09: Populate state-action pair (s_t, a_t)
10: Arbitrarily set $Q(s_t, a_t)$ and solve (10) and observe reward $R(\gamma^*, \cdot)$
11: **If** condition **C3** is true **then**
12: Update system using (36)
13: **Else**
14: Change to cooperative learning
15: **End If**
16: Populate current transition probabilities and observe tuple $(s_t, a_t, p_t, R_t, s_{t+1})$
17: **End For**

Algorithm 2: Proposed Nearest Neighbor Cooperative DRL Algorithm

01: Initialize θ_t with zero-mean normal distribution
02: Initialize number of chosen neighbors, and
03: Construct discrete state space of neighbor SBSs
04: Learn distance metric to the nearest SBS
05: **For** $i = 1 : T$ **do**
06: Learn the optimal Q-function using nearest neighbor regression method
07: Execute steps **06 - 08** to train DNN model
08: Populate state-action pairs (s_t, a_t)
09: Draw action $a_t \approx \pi(\cdot|\mathcal{S})$ to solve (10)
10: Observe the reward $R(\gamma^*, \cdot)$ and generate next state $s_{t+1} \approx p_t(\cdot|s_t, a_t)$
11: **If** condition **C3** is true **then**
12: Update system using (36)
13: **Else**
14: Increase transmission power
15: **End If**
16: Populate current parameters and observe the tuple $(s_t, a_t, p_t, R_t, s_{t+1})$
17: **End For**

accuracy, δ, is used for both proposed algorithms. In the individual learning algorithm that is given in **Algorithm 1** above, the RL agent learns from both positive and negative rewards after executing an action $a \in \mathcal{A}(s)$. For instance, after the initialization of parameters, the next step is to train the DNN model using the input sample of channel gains, i.e., **lines 05 - 07**. This is done in order to come up with a set of possible candidate transmission powers, i.e., the actions. For this task, an efficient stochastic gradient descent (SGD) algorithm [35], which derives the gradient by a running average of its recent magnitude, is considered. The DNN model is updated in order to scale the target variable. In this case, it must be noted that when the scale of the target variable is reduced, the size of the gradient used to update the weights is also reduced, hence a more stable model and training process is realized. As discussed in Section IV-A, with the weights of the DNN represented using probability distributions over possible values of the observable environmental states, $P(w|\mathcal{S})$, the uncertainty in the hidden layers allow for the expression of uncertainty about the outputs [41]. However, it must be observed that for $|a_j| \gg 1$, $\Phi_{sig}(a_j) \approx \Phi_{0,1}(a_j)$, as long as the weights of the network are not regularized. Therefore, a DNN of this depth can be approximated by polynomial networks of depth $\mathcal{O}(\log\log(1/\delta))$, for some fixed accuracy, δ. Similarly, the ReLU function, $\sigma_{relu}(a_j)$ equals to $\sigma_{0,1}(a_j)$ for every $a_j \gg 1$.

1) COMPLEXITY OF THE ACTION SELECTION STRATEGY

With the output of the DNN being the set $\mathcal{A}$ of candidate transmission powers, the Q-learning algorithm is then used to select the best action, i.e., the optimal power. This is an exploration performed using an ϵ-greedy policy, where a greedy action is taken with respect to the estimated Q-function with probability $1 - \epsilon$, and a random one with probability ϵ [36]. A persistent exploration learning policy π is used to store the information about the relatedness of actions in the states in a Q-table. This being an undirected exploration, i.e., it uses only the Q-values, the algorithm has no information about action selection on which it bases its decisions. Since the DNN does not directly provide an estimate of uncertainty, relying on ϵ-greedy exploration results in a low sample efficiency because of the undirected exploration. The action-value function, $Q(s, a)$, is then modeled using Q-learning and iteratively improved by the DNN by minimizing the loss function [42]. This is the action selection step in **line 08** that implements the exploration rule defining which state to go to next. In the individual learning algorithm the agent is only allowed to look for information local to the state of the SBS. This includes the Q-values for all actions, $a \in A(s)$. As stated in [43], the number of steps executed is always bounded by an expression that depends only on the initial and current Q-values. The complexity of action selection which is well elaborated in [44] is $\mathcal{O}\left(\frac{N}{\delta^2(1-\beta^t)}\log^2\frac{N}{\delta(1-\beta^t)}\right)$. However, for the proposed DRL algorithm, an effective finite horizon power control condition is expected to reduce the sample selection complexity to $\frac{1}{(1-\beta^t)}$. Therefore, the expected time and space complexity of action selection is $\mathcal{O}\left(\frac{1}{\delta^2(1-\beta^t)}\log\frac{1}{\delta(1-\beta^t)}\right)$.

2) UPDATING AND REACHING THE REWARD STATE

After the action execution step, state-action pairs are populated, the $Q(s_t, a_t)$ is set and maintained. The value $Q(s_t, a_t)$

is then used to approximate the optimal total reward received when the agent starts in s_t, executes a_t and behaves optimally afterwards. If the congestion condition is true, an update step, **line 12**, adjusts $Q(s_t, a_t)$, and if needed, other information local to the previous state. This means that the one-step look-ahead value $R(s, a)+\beta^t R(s, a)$ is more accurate, and therefore replaces $Q(s_t, a_t)$. An immediate reward $R(s, a) \in \mathcal{R}$ is then obtained, and if the agent starts in $s \in \mathcal{S}$ and executes actions for which it receives immediate reward R_t at time step t, then the total reward that the agent receives over its lifetime for this particular behavior is $R(\gamma^*, \cdot) = \sum_{t=0}^{\infty} \beta^t R_t$. Finally, the current transition probabilities are populated into a tuple $(s_t, a_t, p_t, R_t, s_{t+1})$ as shown in **line 16** of **Algorithm 1**. As the agent approaches the reward state, the number of steps can be exponential in the number of states. The Q-function value of each state-action pair can be augmented with an estimate of its uncertainty to guide exploration, and to achieve faster learning and a higher reward during learning. For example, if the RL strategy uses a zero-initialized Q-learning algorithm, i.e., all Q-values are initially zero, and operates on the reward representation, the first Q-value that changes is the one for the action leading to the reward state. For all the other actions, no information about the topology of the state space is remembered and all Q-values remain zero. For the resource management purpose considered in this work, because the Q-learning algorithm is admissible, the Q-values remain consistent and monotonically decreasing. Due to the monotonic decrease of the Q-values, the sum of the Q-values also decreases with every step, but it is bounded from below such that the algorithm must terminate. Thus, assuming that the state space has no duplicate actions, and the shortest distance between any two states is bounded by $n - 1$, the feasibility result in [46] follows directly. The complexities of the baseline [24] and the proposed algorithms are summarized in TABLE 2 below. Utilizing the invariant and the fact that each of the e different Q-values is bounded by an expression that depends on reward distances to derive a bound on t. From TABLE 2 above, it is evident that the greedy action selection complexity and learning update complexity of the proposed individual learning algorithm are less than those of the baseline algorithm. However, it is clear that as more elements are added to it, i.e., with DRL, the algorithm becomes more computationally complex to implement. However, if the system is no longer able to learn everything from its own observations and experiences, it switches to the nearest neighbor cooperative learning strategy. The dynamics of the nearest neighbor cooperative algorithm may be long in terms of the required operations and memory, with the computational complexity increasing with the increase in the observation space. Locating the nearest SBS according to the Euclidean distance is because the nearest neighbor algorithm critically depends on metric spaces. Thus, $e \leq n^2$ and the worst-case complexity becomes $\mathcal{O}(n^2)$, which is the upper bound on the complexity of the Q-learning algorithm. Therefore, due to the transfer learning scheme, its worst-case complexity is quadratic, i.e., $\mathcal{O}(n^2)$.

TABLE 2. Computational complexity of proposed algorithms compared with the baseline.

Strategy	Algorithm	Action Selection	Learning Update	Overall
Baseline	Q-learning	$\mathcal{O}(n \log^2 n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$
Proposed	Individual	$\mathcal{O}(n \log n)$	$\mathcal{O}(n)$	$\mathcal{O}(n \log n)$
	Cooperative	$\mathcal{O}(n \log n)$	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2)$

V. PERFORMANCE EVALUATION

In this section, the performance results of the proposed algorithms are evaluated via simulations on MATLAB^{TM} software, running on a workstation computer with an Intel i5 core, 3.2 GHz processor. The simulation parameters are tabulated in TABLE 3 below.

TABLE 3. Simulation parameters used for evaluating algorithms.

Parameter	Value
Maximum number of UEs, K	40
UE mobility model	Random waypoint
Packet length, L	5000 bits
Traffic type	Constant bitrate
Time slot duration, Δt	10 ms
Fixed symbol rate, $1/T_s$	500 $\times 10^3$ symbols/sec
Achievable data rate, σ/T_s	bits/sec
Capacity threshold, ψ	90%
Exploration rate, ϵ	0.9
Exploration decay rate	0.995
Learning step sizes, α_t	0.4 - 0.6
Discount factor, β^t	0.98

The performance is evaluated on a 250 $\times$ 250 grid urban environment, with a distance-based path-loss at component carrier frequency of 28 GHz and component system bandwidth of 100 MHz [45]. Here, $M = 10$ randomly deployed SBSs share learning information with each other as a way of controlling their individual congestion levels. The rest of the Here, the SBS transmission power is set at 20 dBm, while the UE are transmitting at 18 dBm with shadow fading (SF) set at 4 dBm. The UE to SBS pathloss is given as: $34.46 + 20\times \log 10(d)$ + SF, and the Gaussian noise power as N_0+ SF as -174 dBm/Hz + SF. The actions are selected randomly according to an ϵ-greedy approach with an exploration decay of 0.995. The optimal actions are learned using step sizes of $\alpha_t = 0.4$ and $\alpha_t = 0.6$, while the reward is discounted with $\beta^t = 0.98$ for better convergence. The analysis is done in three experiments: (i) congestion rate, (ii) achievable bit rate, and (iii) user satisfaction. For each of these experiments, *individual learning*: where SBSs are considered as non-coordinated and independent learners; and *cooperative learning*: where the SBSs are considered in the multi-agent domain whose learning performance is not independent of other SBSs; are evaluated, and their performance is compared with baseline DRL algorithms from [24].

A. EXPERIMENT 1 - CONGESTION RATE

Here, channel utilization is used in conjunction with the utilization thresholds in order to identify the level of congestion

in the serving SBS. The results presented in this section show the congestion rate of the SBS evaluated as a function of the number of admitted UEs for the two different learning schemes, i.e., individual learning and cooperative learning.

1) INDIVIDUAL LEARNING ALGORITHM

The performance of the individual learning algorithm for evaluating the congestion rate when the learning rate is $\alpha_t = 0.4$ is shown in FIGURE 2 below.

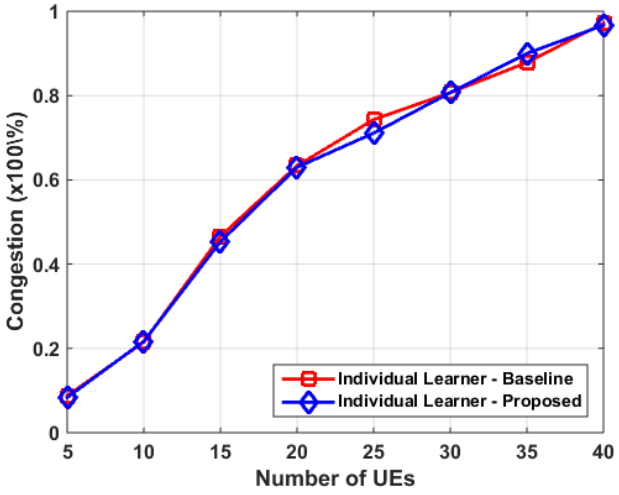

FIGURE 2. Congestion rate at SBS using individual learning with $\alpha_t = 0.4$.

FIGURE 2 above shows that at $\alpha_t = 0.4$, the proposed algorithm and the baseline algorithm exhibit similar performance, with their performances distributed at approximately 8% and 21% for $K = 5$ and $K = 10$, respectively. This means that both algorithms achieve a high level of generalization when the number of admitted UEs is low and there is a 90% probability that the SBS will satisfactorily serve all the UEs. However, when the number of UEs is $10 \leq K \leq 40$, the difference between individual learning and cooperative learning becomes discernible. Their capability for learning and estimating congestion becomes distinctly distinguishable. With the individual learning algorithms, both conventional and proposed learning schemes show similar weaknesses in avoiding congestion.

The performance of the individual learning algorithm for evaluating the congestion rate when the learning rate is increased to $\alpha_t = 0.6$ is shown in FIGURE 3 below.

FIGURE 3 above shows the performance evaluation of the congestion rate at learning rate $\alpha_t = 0.6$ for individual learning. Here, the performance of both the proposed individual learning algorithm and the baseline algorithm is the same for $5 \leq K \leq 25$. In as much as the proposed scheme follows the same performance trajectory with the baseline algorithm, there is 2.2% better performance than the baseline algorithm, and 1.2% better than when the learning rate was $\alpha_t = 0.4$. At $K = 40$, the performance is distributed at approximately 93.2%, which is 3.5% better performance than at $\alpha_t = 0.4$. This means that with $\alpha_t = 0.6$, the SBS has

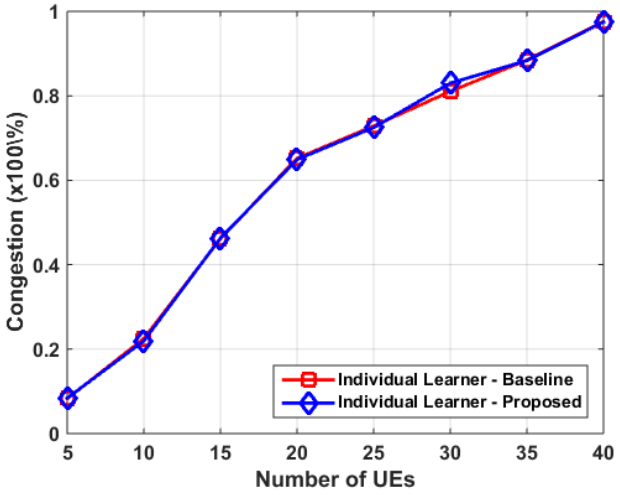

FIGURE 3. Congestion rate at SBS using individual learning with $\alpha_t = 0.6$.

a 3.5% better capability of satisfying all the admitted UEs than when $\alpha_t = 0.4$. Thus, the choice of the learning rate has an effect on network congestion, as was determined by applying different magnitudes of the learning rate in [30]. However, in as much as increasing the learning rate appears to give better statistical performance, it introduces instability in the learning performance, as can be observed at $K = 30$ where the performance of the proposed algorithm overrides the baseline instead of the opposite. Because of the presence of the DNN, a higher learning rate results in learning a set of sub-optimal weights too quickly - hence an unstable training process. This shows that by changing the learning rate, there is a trade-off between better performance and stability of the DRL algorithm.

2) NEAREST NEIGHBOR COOPERATIVE ALGORITHM

In this case, since the selected nearest neighbor knows the trajectory leading to the reward, it is then treated as a mentor that performs transfer learning to this tagged SBS. This assumption is actually consistent with the view of RL as a form of automatic programming [36]. Therefore, in this part, the SBS is assumed to be learning with the assistance of the nearest neighbor for congestion control. The influence of the nearest neighbor was investigated and verified in [24] and [30].

In FIGURE 4 above, the effect of using the cooperative algorithm is noticeable as the performance of the proposed algorithm begins to improve with an increasing number of UEs. The performance of the proposed cooperative algorithm is distributed approximately at 7.1% at $K = 5$, which is 2.2% better than the baseline algorithm. However, in the range $7 < K < 17$, its performance becomes poor, only to retain superiority at $K < 17$. The effect of increasing the learning rate, α_t, under the cooperative framework is investigated in FIGURE 5 below.

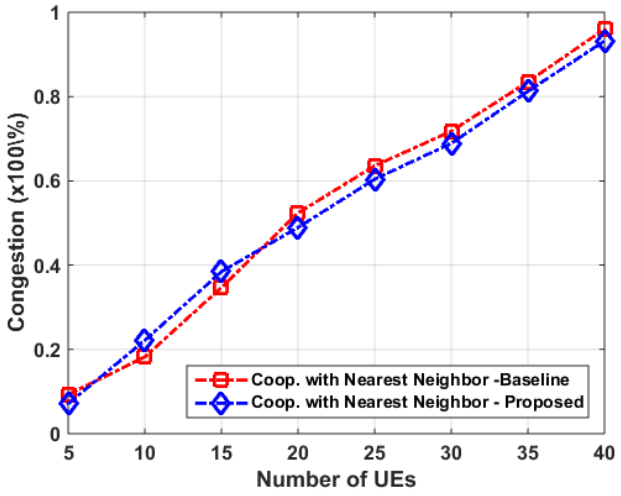

FIGURE 4. Congestion rate with SBS cooperating with nearest neighbour at learning rate $\alpha_t = 0.4$.

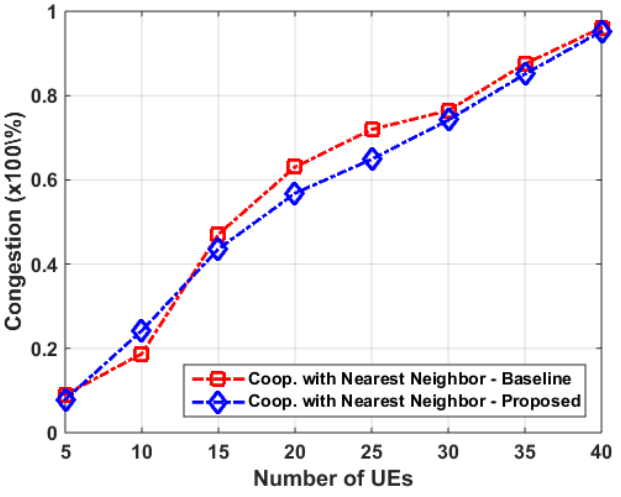

FIGURE 5. Congestion rate with SBS cooperating with nearest neighbour at learning rate $\alpha_t = 0.6$.

The performance of the cooperative algorithm in FIGURE 5 above exhibits similar behavior to that observed in FIGURE 4. The increase in learning rate from $\alpha_t = 0.4$ to $\alpha_t = 0.6$ does not have much effect, which means that the algorithm is insensitive to a change in learning rate. This feature is attractive for future IAB networks where nodes will have the self-reconfiguration capabilities to adjust their parameters in changing environmental conditions.

B. EXPERIMENT 2: THROUGHPUT - ACHIEVABLE BIT RATE

The extension of the model from congestion rate to achievable bit rate is crucial in order to monitor the throughput performance at different levels of congestion. Thus, in this section, the performance of the proposed algorithms in terms of the achievable bit rate is evaluated. The aim was to maximize the achievable bit rate under two constraints, **C1** and **C2**. This means that the central idea is to is to enforce admission control based on the satisfaction rate, $\Omega(\gamma^*, [0, T]) \approx \sigma/T_s$.

1) INDIVIDUAL LEARNING ALGORITHM

The performance of the individual learning algorithm on throughput is evaluated with a learning rate of $\alpha_t = 0.4$ in FIGURE 6 below.

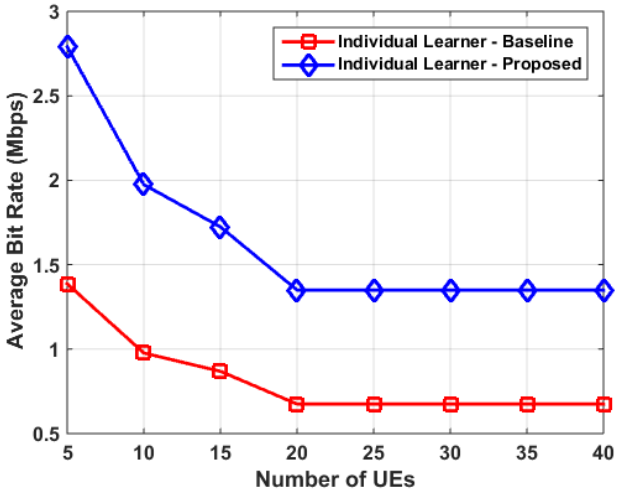

FIGURE 6. Achievable bit rate at SBS using individual learning with a learning rate $\alpha_t = 0.4$.

In FIGURE 6, the throughput performance is evaluated in terms of the achievable bit rate as a function of an increasing number of admitted UEs. With $\alpha_t = 0.4$, the proposed algorithm outperforms the baseline by 46.79% at $K = 5$; however, in the range $20 \geq K \geq 40$, the difference is about 22.50%. It can be observed that the throughput performance of the proposed individual learning algorithm is superior to that of the baseline algorithm. This is evidenced by the fact that the amount of throughput that the proposed algorithm achieves in the range $20 \geq K \geq 40$, i.e., 1.35 Mbps, is what the baseline achieves with $K = 5$. The throughput performance of both methods can be seen to stabilise in the range $20 \geq K \geq 40$. This is because the solution methods aim to keep satisfying the QoS requirements of all UEs until the SBS can no longer admit more UEs. It was observed from the results in [30] that as the number of UEs increases above 40 for the same model, the throughput would significantly drop especially for the cooperative learning scheme, thus indicating a requirement for a change in the bandwidth split or migration of UEs.

In order to further evaluate the performance of this selfish behavior, where SBSs try to increase their throughput independently, the throughput performance for the individual learning algorithm is evaluated with a learning rate of $\alpha_t = 0.6$ as shown in FIGURE 7 below.

In FIGURE 7 above, the performance of the individual learning algorithm is evaluated for $\alpha_t = 0.6$. At $K = 5$, the performance of the proposed algorithm is 46.55% higher, and 22.50% when $K \geq 20$. Similar to Fig. 6, a throughput of 1.35 Mbps is achieved between the $20 \geq K \geq 40$ range, which is the rate achieved by the baseline algorithm at $K = 5$. There is a 0.0113 Mbps decline in the throughput observed after the learning rate is increased, and there is only a 0.24%

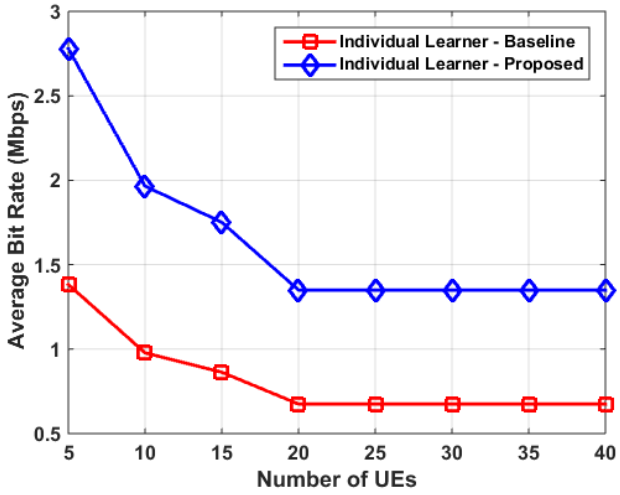

FIGURE 7. Achievable bit rate at SBS using individual learning with $\alpha_t = 0.6$.

difference separating both learning rates at $K = 5$. From this, it can be concluded that there is not much significant sensitivity to the change in learning rate for the individual learning algorithm. However, the consistent throughput performance in the range $20 \geq K \geq 40$ for both learning rates means that the individual learning algorithm is not affected by the congestion rate at the SBS.

2) NEAREST NEIGHBOR COOPERATIVE ALGORITHM

In this part, cooperation between SBSs is employed in order to significantly improve the overall system throughput. The sharing of learned policies in the form of value functions with the nearest neighbor is imposed on the set of states and the system variables defined in the state space. The throughput performance in terms of the achievable bit rate is evaluated using the cooperative algorithm, beginning with $\alpha_t = 0.4$, as shown in FIGURE. 8 below.

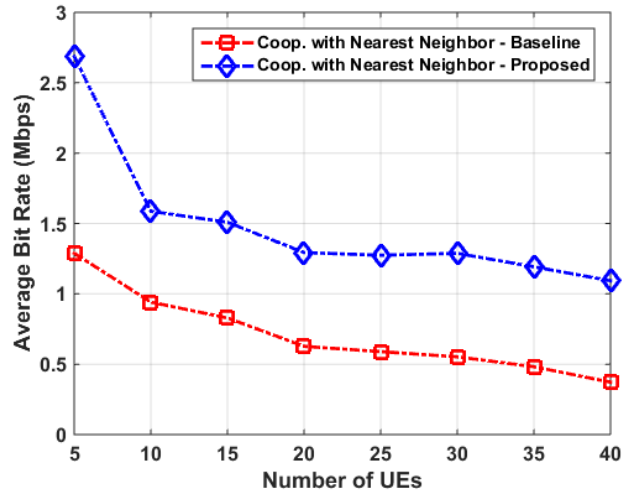

FIGURE 8. Achievable bit rate at SBS cooperative with nearest neighbor at learning rate $\alpha_t = 0.4$.

From FIGURE 8 above, when $K = 5$ the proposed cooperative algorithm displays a 46.56% superiority over the baseline algorithm, with the difference decreasing to 21.52% at K = 10. The achievable bitrate decreases steadily, maintaining almost the same 21.52% difference from the baseline algorithm until $K = 40$. The throughput performance is evaluated for $\alpha_t = 0.6$ in FIGURE 9 below.

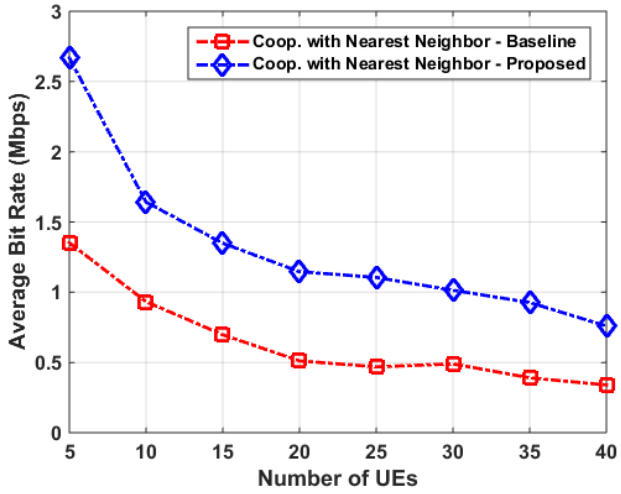

FIGURE 9. Achievable bit rate at SBS cooperating with nearest neighbor at learning rate $\alpha_t = 0.6$.

In FIGURE 9 above, the learning rate $\alpha_t = 0.6$ performs 43.84% better than the baseline algorithm at $K = 5$. There is a noticeable decline of 2.72% in the difference from when $\alpha_t = 0.4$. With the increase in the number of UEs, the performance continues to decline in a noticeably smooth pattern. This means that the cooperative algorithm is sensitive to an increase in the learning rate and an increasing number of UEs on the access network. It can be concluded that as the number of UEs increases, at larger values of $\alpha_t : \alpha_t \rightarrow \beta^t$, the cooperative algorithm prohibits better throughput. As a result, the performance of the nearest neighbor cooperative algorithm is sensitive to the change in the learning rate. These results mean that the throughput performance of the nearest neighbor cooperative algorithm is affected by the congestion rate. This claim is justified by the behavior shown in FIGURE 4 at $K = 17$, and in FIGURE 5 at $K = 13$.

C. EXPERIMENT 3: QUALITY OF EXPERIENCE

After the evaluation of the achievable throughput, the evaluation of user satisfaction is used to qualify the users' perception of the service, that is, the QoE. This is a quality metric that is measured using the mean opinion score (MOS). The system-level parameters related to the throughput discussed above, i.e., packet losses and the delay are used to measure the QoE, using the MOS scale of 1 (worst) - 5 (best). In this experiment, the system utilization is used in conjunction with the throughput to measure the overall QoE of the admitted users. The QoE, a popular factor for measuring the success of multimedia services, is an indicator of user experience and

user satisfaction [47]. The satisfaction of users is computed using the regression of the relationship between the required QoS and the available transmission time, taking into account the congestion rate. It is then measured based on the MOP, with 5 being the maximum score.

1) INDIVIDUAL LEARNING ALGORITHM

User satisfaction is evaluated using the individual learning algorithm for $\alpha_t = 0.4$ and the result is shown in FIGURE. 10 below.

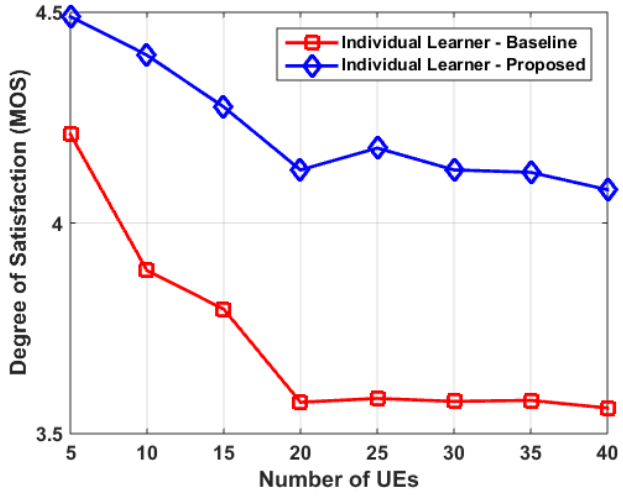

FIGURE 10. UE satisfaction at SBS using individual learning with $\alpha_t = 0.4$.

As observed from the results in FIGURE 10 and FIGURE 11, the satisfaction degree of UEs declines uniformly between $5 \leq K \leq 20$, and tends to exhibit consistency for the remainder of the range. It was observed that there is a significant sensitivity to the change in learning rate for the individual learning scheme. With a learning rate of $\alpha_t = 0.4$, the satisfaction of UEs is as high as 4.5 at $K = 5$, and distributed around 4.1 for the range $20 \leq K \leq 40$. The learning rate is changed to $\alpha_t = 0.6$ in FIGURE 11 below.

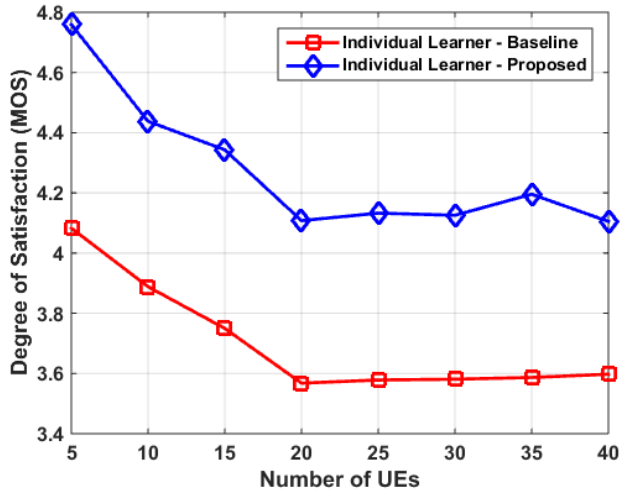

FIGURE 11. UE satisfaction at SBS using individual learning with $\alpha_t = 0.6$.

When $\alpha_t = 0.6$, at $K = 5$ the satisfaction degree is distributed at approximately 4.8, and distributed at approximately 4.1 for the range $20 \leq K \leq 40$.

2) NEAREST NEIGHBOR COOPERATIVE LEARNING ALGORITHM

In this part, the performance of the nearest neighbor cooperative algorithm is evaluated in terms of the satisfaction degree with $\alpha_t = 0.4$; the results are shown in FIGURE 12 below.

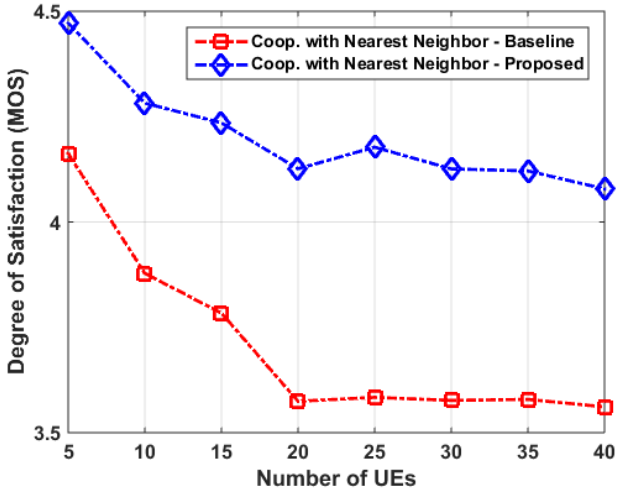

FIGURE 12. UE satisfaction at SBS cooperating with nearest neighbor at learning rate $\alpha_t = 0.4$.

The evaluation of the nearest neighbor cooperative algorithm for $\alpha_t = 0.6$ is shown in FIGURE 13 below.

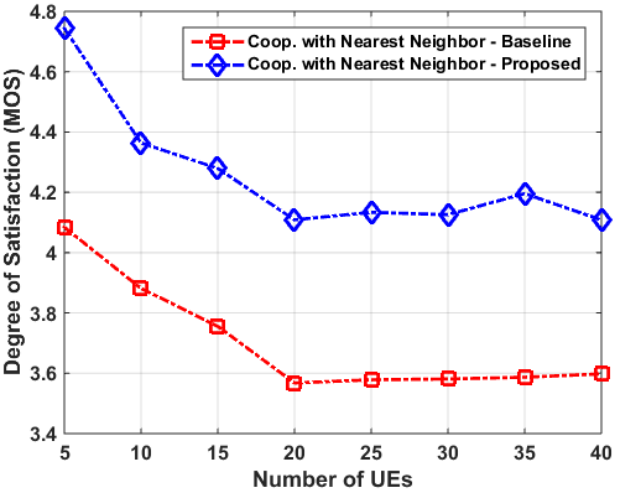

FIGURE 13. UE satisfaction at SBS cooperating with nearest neighbor at learning rate $\alpha_t = 0.6$.

When exploiting the nearest neighbor cooperative learning algorithm, the good capability of offering better QoE similar to the individual learning algorithm means that user satisfaction performance is not sensitive to either the learning

scheme or a change in the learning rate. These results are in agreement with recent findings and supports the results and analysis in [48]. Even though a discount factor that is closer to 1 requires more time to converge, better performance is guaranteed, which is the objective of this work. Another reason for using a higher discounting factor is the advantage that it has in congestion control. Since congestion is incremental, a higher discount on the rewards maximizes generalization and also avoids over-fitting of earlier learning.

VI. CONCLUSION

In this article, a radio resource management solution that aims to avoid congestion on the access side of IAB networks was proposed. In order to provide satisfactory RA for users, the congestion rate of an IAB node is monitored by introducing a transmission buffer. Because of the power consumption issues in the RA framework, the problem was converted into a constrained problem using MDPs and dynamic power management. A DRL algorithm was then proposed, where a DNN was trained for optimal power allocation by initializing a power control parameter, θ_t, with zero-mean normal distribution. The DRL algorithm then adopted its output to learn a policy, π, parameterized by θ_t, to achieve logical allocation of resources by placing more emphasis on congestion control and user satisfaction. The performance of the proposed DRL algorithm was evaluated using two learning schemes - individual learning and nearest neighbor cooperative learning. It was found that the nearest neighbor cooperative learning algorithm is suitable for IAB networks because its throughput has good correlation with the congestion rate. From the algorithmic computational complexity analysis, it is evident that the greedy action selection and learning update complexities of the proposed individual learning algorithm are less compared to the baseline Q-learning algorithm. However, the learning update computational complexity, and consequently the overall complexity of the cooperative learning scheme is the same as that of the baseline algorithm. Power consumption analysis and energy efficiency performance evaluation is considered for future work.

REFERENCES

[1] M. Marchese, A. Moheddine, and F. Patrone, "IoT and UAV integration in 5G hybrid terrestrial-satellite networks," *Sensors*, vol. 19, no. 17, pp. 1–19, Aug. 2019.

[2] A. Osseiran, J. F. Monserrat and P. Marsch, *5G Mobile and Wireless Communications Technology*. Cambridge, U.K.: Cambridge Univ. Press, 2016.

[3] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1136–1160, 3rd Quart., 2013.

[4] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "Networks and devices for the 5G era," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 90–96, Feb. 2014.

[5] J. S. Perez, S. K. Jayaweera, and S. Lane, "Machine learning aided cognitive RAT selection for 5G heterogeneous networks," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, İstanbul, Turkey, Jun. 2017, pp. 1–5.

[6] M. Zorzi, A. Zanella, A. Testolin, M. De F. De Grazia, and M. Zorzi, "Cognition-based networks: A new perspective on network optimization using learning and distributed intelligence," *IEEE Access*, vol. 3, pp. 1512–1530, Aug. 2015.

[7] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Context aware computing for the Internet of Things: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 414–454, 1st Quart., 2014.

[8] S. Singh, M. N. Kulkarni, A. Ghosh, and J. G. Andrews, "Tractable model for rate in self-backhauled millimeter wave cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 10, pp. 2196–2211, Oct. 2015.

[9] M. Polese, M. Giordani, T. Zugno, A. Roy, S. Goyal, D. Castor, and M. Zorzi, "Integrated access and backhaul in 5G mmWave networks: Potential and challenges," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 62–68, Mar. 2020.

[10] S. Jin, J. Liu, X. Leng, and G. Shen, "Self-backhaul method and apparatus in wireless communication networks," U.S. Patent 2007 0 110 005 A1, May 17, 2007.

[11] *What You Need to Know About 5G Wireless Backhaul*, Ceragon, Tel Aviv-Yafo, Israel, 2016.

[12] A. Betzler, D. Camps-Mur, E. Garcia-Villegas, I. Demirkol, and J. J. Aleixendri, "SODALITE: SDN wireless backhauling for dense 4G/5G small cell networks," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1709–1723, Dec. 2019.

[13] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454–2467, Sep. 2019.

[14] K. Xiong, S. S. R. Adolphe, G. O. Boateng, G. Liu, and G. Sun, "Dynamic resource provisioning and resource customization for mixed traffics in virtualized radio access network," *IEEE Access*, vol. 7, pp. 115440–115453, Aug. 2019.

[15] F. J. Martin-Vega, M. Di Renzo, M. C. Aguayo-Torres, G. Gomez, and T. Q. Duong, "Stochastic geometry modeling and analysis of backhaul-constrained hyper-dense heterogeneous cellular networks," in *Proc. 17th Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2015, pp. 1–4.

[16] A. Sharma, R. K. Ganti, and J. K. Milleth, "Joint backhaul-access analysis of full duplex self-backhauling heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1727–1740, Mar. 2017.

[17] E. Turgut and M. C. Gursoy, "Coverage in heterogeneous downlink millimeter wave cellular networks," *IEEE Trans. Commun.*, vol. 65, no. 10, pp. 4463–4477, Oct. 2017.

[18] C. Saha, M. Afshang, and H. S. Dhillon, "Bandwidth partitioning and downlink analysis in millimeter wave integrated access and backhaul for 5G," *IEEE Trans. Wireless Commun.*, vol. 17, no. 12, pp. 8195–8210, Dec. 2018.

[19] C. Saha and H. Dhillon, "Millimeter wave integrated access and backhaul in 5G: Performance analysis and design insights," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 12, pp. 2669–2684, Dec. 2019.

[20] R. Taori and A. Sridharan, "Point-to-multipoint in-band mmWave backhaul for 5G networks," *IEEE Commun. Mag.*, vol. 53, no. 1, pp. 195–201, Jan. 2015.

[21] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1823–1834, May 2010.

[22] J. He, J. Peng, F. Jiang, G. Qin, and W. Liu, "A distributed Q learning spectrum decision scheme for cognitive radio sensor network," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 5, May 2015, Art. no. 301317.

[23] H. Jiang, H. He, L. Liu, and Y. Yi, "Q-learning for non-cooperative channel access game of cognitive radio networks," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2018, pp. 1–7.

[24] W. Lei, Y. Ye, and M. Xiao, "Deep reinforcement learning-based spectrum allocation in integrated access and backhaul networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 3, pp. 970–979, May 2020.

[25] *Study on Integrated Access and Backhaul*, document 38.874, 3GPP, Release 16, 2018.

[26] M. B. A. Sadi and A. Nadia, "Call admission scheme for multidimensional traffic assuming finite handoff user," *J. Comput. Netw. Commun.*, vol. 2017, pp. 1–5, Mar. 2017.

[27] M. Simsek, D. Zhang, D. Öhmann, M. Matthé, and G. P. Fettweis, "On the flexibility and autonomy of 5G wireless networks," *IEEE Access*, vol. 5, pp. 22823–22835, Jun. 2017.

[28] A. Chydzinski and B. Adamczyk, "Queues with the dropping function and general service time," *PLoS ONE*, vol. 14, no. 7, pp. 1–21, Jul. 2019.

[29] W.-W. Fang, J.-M. Chen, L. Shu, T.-S. Chu, and D.-P. Qian, "Congestion avoidance, detection and alleviation in wireless sensor networks," *J. Zhejiang Univ. Sci. C*, vol. 11, no. 1, pp. 63–73, Jan. 2010.

[30] M. M. Sande, M. C. Hlophe, and B. T. Maharaj, "Instantaneous load-based user association in multi-hop IAB networks using reinforcement learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
[31] M. C. Hlophe and B. T. Maharaj, "QoS provisioning and energy saving scheme for distributed cognitive radio networks using deep learning," *J. Commun. Netw.*, vol. 22, no. 3, pp. 185–204, Jun. 2020.
[32] V. Pla, A. S. Alfa, J. Martinez-Bauset, and V. Casares-Giner, "Discrete-time analysis of cognitive radio networks with nonsaturated source of secondary users," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–12, Jan. 2019.
[33] A. Slalmi, H. Chaibi, A. Chehri, R. Saadane, and G. Jeon, "Toward 6G: Understanding network requirements and key performance indicators," *Trans. Emerg. Telecommun. Technol.*, vol. 32, no. 3, p. e4201, Mar. 2021.
[34] N. Mastronarde and M. van der Schaar, "Joint physical-layer and system-level power management for delay-sensitive wireless communications," *IEEE Trans. Mobile Comput.*, vol. 12, no. 4, pp. 694–709, Apr. 2013.
[35] M. C. Hlophe and S. B. T. Maharaj, "Spectrum occupancy reconstruction in distributed cognitive radio networks using deep learning," *IEEE Access*, vol. 7, pp. 14294–14307, Jan. 2019.
[36] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.
[37] R. Cimurs, J. H. Lee, and I. H. Suh, "Goal-oriented obstacle avoidance with deep reinforcement learning in continuous action space," *Electronics*, vol. 9, no. 3, pp. 1–16, Mar. 2020.
[38] A. Choudhary. *A Hands-On Introduction to Deep Q-Learning Using OpenAI Gym in Python*. Accessed: Jun. 1, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2019/04/introduction-deep-qlearning-python/
[39] M. C. Hlophe and B. T. Maharaj, "Secondary user experience-oriented resource allocation in AI-empowered cognitive radio networks using deep neuroevolution," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC-Spring)*, May 2020, pp. 1–5.
[40] M. Carter and B. van Brunt, *The Lebesgue-Stieltjes Integral: A Practical Introduction*. New York, NY, USA: Springer, 2002.
[41] H. Cuayáhuitl, S. Keizer, and O. Lemon, "Strategic dialogue management via deep reinforcement learning," Nov. 2015, pp. 1–10, *arXiv:1511.08099*. [Online]. Available: http://arxiv.org/abs/1511.08099
[42] C. Ding, S. Liao, Y. Wang, Z. Li, N. Liu, Y. Zhuo, C. Wang, X. Qian, Y. Bai, G. Yuan, X. Ma, Y. Zhang, J. Tang, Q. Qiu, X. Lin, and B. Yuan, "CirCNN: Accelerating and compressing deep neural networks using block-circulant weight matrices," in *Proc. 50th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Oct. 2017, pp. 395–408.
[43] S. Koenig and R. G. Simmons, "Complexity analysis of real-time reinforcement learning," in *Proc. AAAI*, 1993, pp. 99–107.
[44] S. M. Kakade, "On the sample complexity of reinforcement learning," Ph.D. dissertation, Gatsby Comput. Neurosci. Unit, Univ. College London, London, U.K., 2003.
[45] *Technical Specification Group Services and System Aspects*, document TR 21.916, 3GPP, Release 15, Oct. 2018.
[46] B. Price and C. Boutilier, "Accelerating reinforcement learning through implicit imitation," *J. Artif. Intell. Res.*, vol. 19, pp. 569–629, Dec. 2003.
[47] L. Pierucci, "The quality of experience perspective toward 5G technology," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 10–16, Aug. 2015.
[48] M. Botvinick, S. Ritter, J. X. Wang, ZebKurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends Cognit. Sci.*, vol. 23, no. 5, pp. 408–422, May 2019.

MALCOLM M. SANDE (Graduate Student Member, IEEE) received the bachelor's and master's degrees in electronic engineering from the University of Pretoria, in 2014 and 2018, respectively, where he is currently pursuing the Ph.D. degree in wireless communications, under the Sentech Chair in Broadband Wireless Multimedia Communications (BWMC) Research Group. His research interests include the application of machine learning techniques in mobile and wireless communications, with a particular interest in radio spectrum management.

MDUDUZI C. HLOPHE (Member, IEEE) received the Ph.D. degree in electronic engineering in the area of wireless communications from the University of Pretoria, South Africa, in 2020. He is currently a Postdoctoral Fellow with Broadband Wireless Multimedia Communications (BWMC), Department of Electrical, Electronic and Computer Engineering, University of Pretoria. His research interests include mathematical modeling of multivariate statistics, classification methods, knowledge discovery, reasoning with uncertainty and inference, and predictive analytics and inference with applications in wireless communications, finance, health, and robotics.

BODHASWAR T. MAHARAJ (Senior Member, IEEE) received the Ph.D. degree in engineering in the area of wireless communications from the University of Pretoria. He is a Full Professor and currently holds the research position of Sentech Chair with Broadband Wireless Multimedia Communications (BWMC), Department of Electrical, Electronic and Computer Engineering, University of Pretoria. His research interests include OFDM-MIMO systems, massive MIMO, cognitive radio resource allocation, and 5G cognitive radio sensor networks.