Received 13 January 2023, accepted 26 January 2023, date of publication 9 February 2023, date of current version 15 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3243519

RESEARCH ARTICLE

A Backhaul Adaptation Scheme for IAB Networks Using Deep Reinforcement Learning With Recursive Discrete Choice Model

MALCOLM M. SANDE, (Student Member, IEEE), MDUDUZI C. HLOPHE, (Member, IEEE), AND BODHASWAR T. SUNIL MAHARAJ, (Senior Member, IEEE)

Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, 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 Challenges such as backhaul availability and backhaul scalability have continued to outweigh the progress of integrated access and backhaul (IAB) networks that enable multi-hop backhauling in 5G networks. These challenges, which are predominant in poor wireless channel conditions such as foliage, may lead to high energy consumption and packet losses. It is essential that the IAB topology enables efficient traffic flow by minimizing congestion and increasing robustness to backhaul failure. This article proposes a backhaul adaptation scheme that is controlled by the load on the access side of the network. The routing problem is formulated as a constrained Markov decision process and solved using a dual decomposition approach due to the existence of explicit and implicit constraints. A deep reinforcement learning (DRL) strategy that takes advantage of a recursive discrete choice model (RDCM) was proposed and implemented in a knowledge-defined networking architecture of an IAB network. The incorporation of the RDCM was shown to improve robustness to backhaul failure in IAB networks. The performance of the proposed algorithm was compared to that of conventional DRL, i.e., without RDCM, and generative model-based learning (GMBL) algorithms. The simulation results of the proposed approach reveal risk perception by introducing certain biases on alternative choices and the results showed that the proposed algorithm provides better throughput and delay performance over the two baselines.

INDEX TERMS Backhaul, choice aversion, constrained Markov decision process, deep reinforcement learning, integrated access and backhaul, recursive discrete choice model, routing.

I. INTRODUCTION

The fifth generation (5G) new radio (NR) technology provides the foundation for future mobile and wireless communications by supporting new types of applications and flexible spectrum usage [1]. The densified small cell architecture of 5G networks makes it labour intensive and costly for mobile network operators (MNOs) to provide fiber backhaul to every access point (AP) in the network [2]. Integrated access and backhaul (IAB) is a key technology enabler for 5G NR that alleviates this challenge by leveraging the availability of large amounts of spectrum in millimeter wave (mm-wave) frequencies to enable the wireless spectrum to be shared between access and backhaul [3]. The IAB technology was standardized for 3GPP release 15 [4], and it is envisaged to be more financially successful than LTE relaying [5]. In an IAB network, there is a wireless backhaul link between secondary base station (SBSs) and a main base station (MBS), which is typically connected to the core network via fiber backhaul. The wireless backhaul links between an SBS and the MBS can be a single direct link or over multiple hops through other SBSs. IAB networks benefit from the usage of high-frequency bands, which are capable of having large transmission bandwidth that is feasible without any considerable performance sacrifice.

In IAB networks, backhaul traffic is routed in an SBS-to-SBS fashion until it reaches the MBS. This brings about the discussion on the need for efficient backhaul routing in

The associate editor coordinating the review of this manuscript and approving it for publication was Petros Nicopolitidis.

IAB networks, which should consider environmental context such as SBS load and bandwidth allocation. As MNOs move from initial 5G market launches to extending their 5G network capacity, they face the challenge of securing high bandwidth backhaul solutions to the 5G sites in a fast and cost-effective manner [3], [6]. Although mm-wave links allow high-throughput wireless transmissions, they are vulnerable to blockage from moving objects such as vehicles, seasonal changes such as foliage, as well as infrastructure changes [7]. Thus, from a reliability perspective, it is important to ensure that each IAB node can continually provide coverage and end-user service even when the active backhaul routes are temporarily unavailable. In order to autonomously reconfigure the backhaul network without service disruption and packet losses during reconfiguration, the 3GPP has standardized the use of topology adaptation. IAB topology adaptation can be a result of integration of a new IAB node into an existing topology, detachment/release of an IAB node from an existing topology, detection of backhaul link overload, deterioration of backhaul link quality, link failure, or other such events. Following is a literature review on some of the recent research works that have used artificial intelligence (AI) strategies for problems in IAB networks.

A. RELATED RESEARCH WORKS

The application of ML techniques in traffic engineering is not new, more especially the supervised learning techniques. Supervised learning models for directly learning paths for high-throughput dynamic packet routing have been proposed in [8] and [9]. For instance, the approach in [9] was used to predict future traffic, then optimize the routing plan using the predicted values. However, simulation results showed that this approach might be ineffective. On the other hand, the approach used in [8] assumes a central controller to avoid congestion, which uses information gathered from the whole network to train a different model for each source and destination pair in the network. The solution for the congestion optimization problem is then provided by a heuristic algorithm. The main challenge that these aforementioned protocols addresses is deciding the best path to be taken by traffic from its source to the destination, under certain constraints.

In addition to supervised learning approaches, reinforcement learning (RL) and deep reinforcement learning (DRL) strategies have also been proposed to solve routing problems. With the view that the conventional routing algorithms do not consider the network data history such as overloaded routes and route failure, the authors in [10] used the advantages of network data to present a RL-based routing strategy. Since RL-based routing algorithms require additional control message headers, the authors addressed this by proposing an enhanced protocol named enhanced RL routing protocol (e-RLRP). The e-RLRP scheme aimed to reduce the network overheads by implementing different network scenarios, where the number of nodes, routes, traffic flows and degree of mobility were varied. The performance of the e-RLRP scheme is compared to that of the optimized link state routing, BATMAN, and RLRP protocols, and the experimental results showed that the e-RLRP protocol provides reduced network overhead in most network scenarios compared to all the other protocols.

In another contribution, the authors in [11] proposed a method to investigate multi-hop scheduling in self-backhauled mm-wave networks. Here, the authors addressed the challenge of selecting the best routes and how to allocate rates to the links subject to latency constraints. In their design, they factored in channel variations and network dynamics that are specific to mm-wave frequencies, and they formulated a network utility maximization problem subject to a bounded delay constraint and network stability. The problem was decoupled into two: (i) path selection and (ii) rate allocation, where learning the best paths was performed using RL, and rate allocation was solved using successive convex approximation. The results of this approach showed that it achieved a guaranteed communication reliability of 99.9999%, and latency reduction of 50.64% and 92.9% when compared with two baselines, respectively. On the other hand, the authors in [12] developed a DRL-based framework to solve the spectrum allocation problem for an IAB architecture with large scale deployment in a dynamic environment. The available spectrum is divided into several orthogonal sub-channels, and the MBS and all IAB nodes have the same spectrum resource for allocation. A spectrum allocation problem was formulated as a mix-integer and non-linear programming problem with the goal of maximizing the sum log-rate of all user equipment (UE) groups. The problem could not be handled when the IAB network became large and time-varying. A DRL strategy was then incorporated in the form of an actor-critic spectrum allocation scheme. Here, deep neural networks (DNNs) were used to achieve real-time spectrum allocation in different scenarios, and the evaluation results were better than some baseline allocation policies.

A novel scheme for jointly allocating spectrum and transmission power for both access and backhaul links for SBSs and MBSs was proposed in [13]. Here, the authors formulated the spectrum allocation and power management problem as a mix-integer and non-linear programming problem, with the objective of maximizing the downlink data rate. A double deep Q-learning network approach was then proposed to achieve an efficient policy learning for joint spectrum allocation and power management, to obtain a scheme named SAPM-DDQN. The proposed SAPM-DDQN does not require any prior information from other units for optimization, which is suitable for practical deployment. Simulation results showed the effectiveness of the proposed scheme for joint spectrum allocation and power management.

B. RESEARCH MOTIVATION

Due to highly dense device connectivity in urban environments, especially during peak hours, network management is becoming more complex. Optimal scheduling in dynamic mm-wave network environments is difficult and relatively time-consuming to perform on-the-fly. In as much as the design of new networks must not lose the features that made them successful; it must be open to new applications, such as being able to adapt to new protocols. This means that new techniques that combine these virtues to new protocols should be devised. From the literature, it has been seen that DRL-based solutions provide a better action-selection strategy that incorporates the dynamic load among APs in IAB networks, compared to conventional Q-learning methods. However, majority of research contributions in terms of resource management problems in IAB networks sought to find an optimal way to allocate a fixed demand for resources from UEs, whose performance degrades with increasing congestion. These approaches usually overlook the fundamental problem related to the features of each application, i.e., the intrinsic coupling of the cost and the demand for network resources. This coupling allows the demand to vary with congestion, thus leading to the ''Tragedy of the Commons'' [14], which is the severe inefficiency caused by the over-consumption of transmission resources. The aim is usually to find the path with the lowest cost according to a defined metric [15].

The most common metric used by routing protocols in literature is the hop count, where the cost of a path is defined as the sum of the number of hops between source and destination. This means that allocating resources to activities such as route request and route exploration/exploitation should be as best as possible. Most optimization problems are formulated as mixed-integer non-linear programming problems in order to reduce the energy consumption costs. Usually, this approach is NP-hard, and evolutionary games are often introduced to deal with their complexities. The cost perspective and computation of the cost function in IAB networks should be defined in terms of the cost of maintaining the required performance levels at each node of the network. Conventional cost functions are either empirical or heuristic. Among all the available cost functions for application-level multicast routing, neither of them has clearly defined derivations. In most of these prior works, the presentations of the routing algorithm do not address how the link cost function should be defined in order to efficiently allocate resources throughout the network. This again raises needs for a new multi-variable cost function. Consequently, many solutions have been developed to solve this problem, and most of them leverage perceptron convergence in neural network (NN) algorithms. The possibility of synthesizing NNs from examples of their input/output behavior is a central motivating factor towards addressing the tragedy of commons in this field.

C. RESEARCH QUESTIONS AND SUMMARY OF CONTRIBUTIONS

In this article, smart ways in which robust and efficient backhauling can be achieved are sought. In doing so, the main questions to ask are: How can a system learn how to handle varying backhaul packet arrival rates without compromising the access quality of service (QoS)? How much of a role do transmission delays and buffer size play in the power management and rate allocation in IAB networks? Can machine learning (ML) techniques be leveraged to improve the energy efficiency and throughput performance in constrained IAB networks? In order to answer these questions, this article examines the effect of packet arrivals on backhaul routing performance metrics in IAB networks while considering latency requirements and buffer size limitations. The contributions of this article are summarized as follows:

1) KNOWLEDGE-DEFINED NETWORKING

Since in IAB networks, part of the radio spectrum is used for backhaul connection, each node must perform dynamic bandwidth reservation in a distributed manner. A knowledge defined networking (KDN) scheme was proposed as an architecture for network monitoring and the bandwidth reservation procedure was carried out in the medium access control (MAC) layer to make the reservation process very rapid. In this case, the proposed system is divided into two subsystems, i.e., (i) the data plane, and (ii) the knowledge plane.

- The **data plane** includes support for distributed traffic admission control (DTAC), any node in the IAB network allocates bandwidth resources for traffic flows in a distributed manner. Assuming that all IAB nodes have the same point of view of the network and employ the same algorithm, the network topology was modeled as a probabilistic graph. Then, the RA problem was formulated as a non-convex programming problem with the objective of maximizing the overall backhaul capacity, subject to a flexible range extension to ensure that the QoS requirements of access users are considered, satisfied, and maintained.
- The **knowledge plane** uses the probabilistic graph model to estimate the Q values and calculate the maximum bound latency, i.e., a process of learning network information from distributed network states is developed using Q-learning. This is a transfer learning procedure for sharing information with nearest neighbors using the forward-backward exploration technique. Here, a performance prediction scheme that uses the principles of DRL strategies was proposed in order to handle the complexity of the IAB network, as well as assessing the latency upper bound and effective throughput.

2) ADDRESSING THE TRAGEDY OF COMMONS

- A QoS-aware routing optimization scenario is presented using a rigorous and unified framework based on constrained Markov decision processes (MDPs), which details the reward and cost functions using implicit and explicit constraints. The basic idea behind this is to be able to simultaneously utilize physical-centric and system-level techniques to achieve maximum throughput and minimum possible delays and power consumption. A crucial cognitive function, where the learning process applies prediction error to adjust future predictions, was incorporated into the DRL strategy. Thus, the DRL strategy was used together with a recursive discrete choice model (RDCM) in the evaluation of route choices in the presence of stochastic traffic and channel conditions. Regret learning, which exploits historical information about channel and queue states in selecting the optimal route, was leveraged with conditions of choice aversion. Then, a DRL strategy was used to aggregate link states on paths in a flexible architecture that represents a source-destination routing scheme.

- A multi-dimensional matrix format is presented to embed the topological and link reliability information of the IAB network. Incorporating attributes such as traffic arrival distribution, channel state, buffer occupancy status, and power management states into one expression is difficult for classical techniques. Here, the mathematical models of queues with deadlines and rewards are used to describe the attributes of the system. Then, in order to gain the best QoS of the IAB network, a max-weight algorithm was used together with back-pressure routing in order to handle the routing aspect of the IAB network. Lastly, taking both into full account both explicit and implicit constraints and several QoS parameters simultaneously, a multi-variable goodput distribution was used to formulate the cost function by employing a post-decision state learning strategy to deal with the known and unknown components of the system. Compared with the conventional DRL algorithm, the improved DRL framework can effectively make better routing decisions and achieves better routing delays.

D. ARTICLE OUTLINE AND NOTATIONS

The remainder of this paper is organized as follows: Section II discusses the proposed system-level model of the IAB network. Section III presents the mathematical formulation of the problem as well as the optimization objective. Section IV discusses the proposed DRL strategy that applies a RDCM. In Section V, the proposed algorithm is discussed in detail, and its computational complexity is compared with the baselines. Section VI presents the performance evaluation of the proposed algorithm in comparison to the baseline approaches using simulation results. Ultimately, Section VII gives the concluding remarks of the article. The notations used in this article, together with their descriptions are tabulated in TABLE 1.

II. PROPOSED SYSTEM MODEL

Consider the uplink (UL) transmission of a two-tier multi-hop IAB network consisting of one IAB donor, a set of IAB nodes, and user equipments (UEs). The IAB nodes serve UEs and are connected to each other via wireless backhaul, while the IAB donor is connected to the core network via fiber backhaul and capable of serving access UEs and backhaul traffic. The proposed IAB network model is shown in Fig. 1 below.

TABLE 1. Notations and Descriptions.

Notation	Description
$\mathcal{G}(\mathcal{V},\mathcal{E})$	Graph $\mathcal{G}$ with node set $\mathcal{V}$, and edge set, $\mathcal{E}$
$v \in \mathcal{V}; e \in \mathcal{E}$	Node v and graph edge e
$\mathcal{N}$	Set of IAB nodes, synonymous with $\mathcal{V}$
$\mathcal{K}$	Set of UEs associated with IAB nodes
$\mathcal{M} \in \mathcal{N} \cup \mathcal{K}$	Set of all nodes, i.e., IAB nodes and UEs
$\mathcal{F}$	A finite set of traffic flows
Q_{th}	Maximum queue length or threshold
q_f^i	Evolution of queue length, i.e., buffer queue state
$e_{i,j}$	A communication link between two nodes i and j
$d_{n,k}$	Eucledian distance between UE k and SBS n
p_k	The transmission power of the k-th UE
$g_{n,k}$	Channel power gain between k-th UE and n-th SBS
$\gamma_{i,j}^m$	SINR of the backhaul link between SBSs i and j
$p_{i,j}^f$	Transmit power from node i to node j
$r_{i,j}^f$	Transmission rate between node i and node j
$\bar{r}$	The average backhaul transmission rate
C_e	The capacity of edge $e \in \mathcal{E}$
$\bar{D}(t)$	The average cost of delayed packets
λ_f	Packet arrival rate at flow f
ζ_f	The maximum delay constraint for each flow
R_f	The discounted reward for a packet in flow f
κ_e	The Lagrange multiplier for edge e
π	The optimal policy

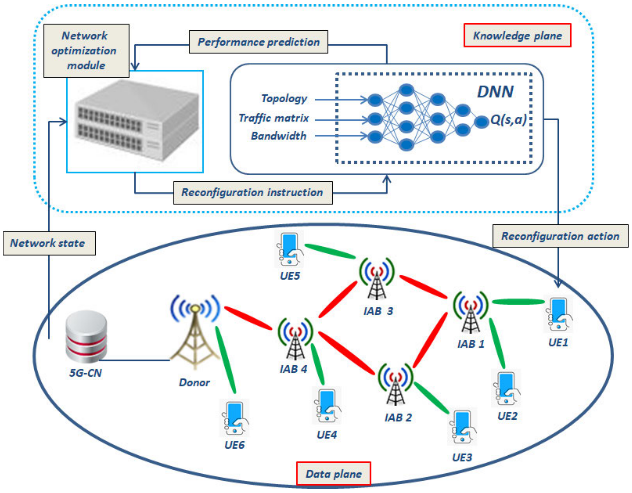


FIGURE 1. IAB network model setup in the 5G standalone deployment scenario.

As shown in Fig. 1 above, $\mathcal{N} = \{0, 1, 2, \cdots, N\}$ access points (APs) are distributed according to a Poisson point process, where n_0 is the IAB donor and the rest are IAB nodes [16]. The donor node together with the IAB nodes are assumed to be equipped with multiple antennas such that they operate in full-duplex mode, i.e., transmitting and receiving signals simultaneously. In line with KDN as part of the 5G network requirements, an IAB network that can assemble itself given high-level instructions, reassemble itself if the requirements change, and autonomously reconfigure itself in the event of an outage, two separate - but communicating planes are proposed, i.e., data plane, and knowledge plane.

A. THE DATA PLANE

In order to allow traffic flows to be scheduled on multiple links, the topology of the IAB network is modeled using

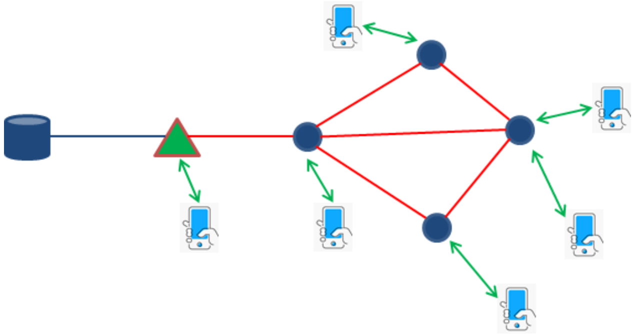


FIGURE 2. A graph model of the data plane of an IAB network.

an undirected graph. Thus, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes (synonymous to the set $\mathcal{N}$), and $\mathcal{E}$ be the set of edges or links, i.e., $\mathcal{E} = \{(i,j)|i \in \mathcal{N}, j \in \mathcal{M}\}$, is defined. Here, i and j represent the indices of the transmitting and receiving nodes, such that $e_{i,j}$ represents the edge/link between the two communicating nodes, and $\mathcal{M} = \mathcal{N} \cup \mathcal{K}$ is the set of all nodes. As shown in Fig. 1 above, the data plane is where the nodes, UEs, and the actual communication channels are found, and it is also where all the signalling and data handling occurs. The data plane of the considered IAB network can be represented using a toy model example as shown in Fig. 2 below [17].

Let $|\mathcal{V}|$ and $|\mathcal{E}|$ denote the cardinality of the node and edge sets, respectively, and $\mathcal{E}_i^+$ represent the set of outgoing links from node i. In addition, let a transmission decision be denoted by (i,j), $\forall i, j \in \mathcal{E}$, while a decision not to transmit be denoted by a loop, i.e., (i,i), $\forall i \in \mathcal{V}$. Then, let the set $\mathcal{F} = \{1, 2, \cdots, F\}$ represent the finite number of traffic flows. Each flow is assumed to have the attributes determining the source and destination nodes. Since in a graph tree, a source node (or a child node or a transmitter) is defined using index i and the destination node (or parent node or receiver) by index j, $\{(i,j)|j = par(i)\}$ describes the parent-to-child relationship in the IAB network. The backhaul links between IAB nodes and their immediate neighbors, up to the destination, are modeled as edges, $e \in \mathcal{E}$, such that when a route request message from the i-th node reaches the parent node j, the communication link is represented as $e_{i,j}$. Let $\mathcal{P}_i$ denote the $i \rightarrow j$ paths of graph $\mathcal{G}$ such that $\mathcal{P} = \cup_i \mathcal{P}_i$ is a set containing the all the paths.

B. THE KNOWLEDGE PLANE

The knowledge plane is a distributed construct within the network that gathers, aggregates, and manages information about network behavior and operation, with a goal of enlarging the view of what constitutes the network [18]. Similar in operation to the control plane, its task is to draw the network topology and handle all the functions and processes that determine which routes to be taken by packets. Therefore, the proposed architecture for this plane exploits the Q-learning technique, with the assumption that all the nodes have the same point of view of the network and run the same algorithm. The exploration of the Q-learning technique on this plane is shown in Fig. 3 below:

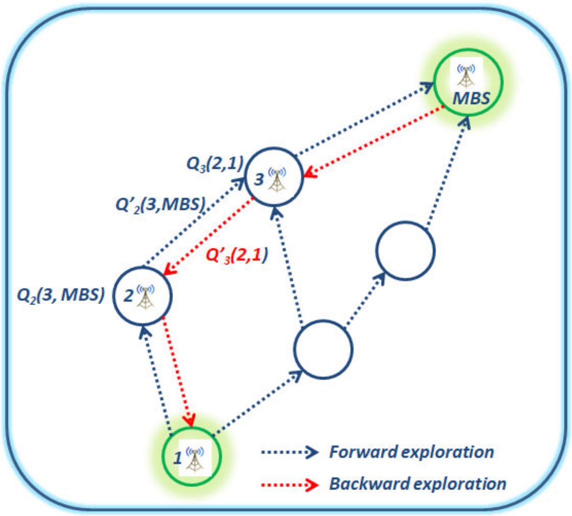


FIGURE 3. The simplified graph model of an IAB environment illustrating the learning framework.

As part of the framework shown in FIGURE 3 above, support for DTAC procedures is incorporated using the max-weight scheme, which frequently tracks node congestion by checking buffer queue occupancy status. This process is a way of evaluating the reliability of every link of nodes that communicate with one another using the backward and forward exploration technique. The ability of the knowledge plane to model the graph-based information of the data plane is made possible through the network optimizing module, which is an OpenFlow device containing the flow table. The implementation of this idea was inspired by the behavioral game discussed in [19].

1) ROUTE ESTABLISHMENT

The process illustrated in FIGURE 3 proceeds as outlined in the following example.

Example 1: Let the route established use the process illustrated in Fig. 3, where the transmission passes through the nodes i, j, and j' towards n_0, such that $Q_j(j', n_0)$ is the time that a node j estimates it takes to deliver a packet bound for n_0 via j'. This time estimate includes the time that the packet will spend in the queue while being buffered at node j, i.e., the holding time. After node j has sent the packet to node j', it immediately receives the estimate of the remaining time for it to reach the destination from node j'. It must be noted that each node in the network maintains information about the Q values for each of the possible next hops. This information represents the delivery time for the packets to reach the MBS. An update regarding the present Q value of each node is sent to the previous node in a process called backward exploration. In order to keep the Q value estimates as accurate as possible, and to also reflect the changes in the state of the network, the estimates need to be updated with minimum possible overhead [17]. Thus, as soon as node j sends a packet $P(i, n_0)$ destined for the MBS to one of the neighboring nodes j', node j' sends its best estimate $Q_{j'}(z, n_0)$ for the destination back to node j, where z is a donor node.

2) COMPUTATION OF BOUNDS

It is believed that the maximum bound on the latency can be calculated using the maximum buffer occupancy of each node and the egress link rate [20], where the queue lengths are used in computing the upper bound of latency in terms of the number of nodes deployed to relay traffic to n_0. Upon receiving the estimate, $Q_{j'}(z, n_0)$, node j computes the new estimate using the exploration of Q values. This process is known as the forward and backward exploration, since it involves updating the Q values of the sending node j using the information obtained from the receiving node j'. With every hop of the packet $P(i, n_0)$, only one Q value is updated, i.e., when node j sends the packet, $P(i, n_0)$, to one of its neighbors, e.g., j', the packet can take along information about the Q values of node j. When node j' receives this packet, it can use this information in updating its Q values pertaining to its neighbor, i.e., node j. Then, when the node j' makes a decision, it uses these updated Q values for node j, then the Q value updates in backward exploration.

III. MATHEMATICAL PROBLEM FORMULATION

Considering that the model described in Section II above is time-slotted with discrete time steps t, the following assumptions are made: (i) the traffic arrival rate, $\lambda_j^f(t)$, at each node queue is approximated by a Poisson process; (ii) the packet lengths are approximated by an exponential distribution, (iii) the traffic arrival distribution, $p^{\lambda}(\lambda)$, is unknown; (iv) a wireless transmission card of each node consists of a transmission buffer that can hold a maximum of $\mathcal{Q}$ packets, whose average queue length, $\bar{q}$, can be explained using Little's theorem [21]. Then, using the number of arrivals and the transmission rate, the evolution of the queue in the transmission buffer can be represented using the dynamic update equation which is elaborated in [22], expressed as follows:

$$q_i^f(t+1) = \left[q_i^f(t) - \sum_{\forall f \in \mathcal{F}} r_{i,j}^f(t), 0 \right]^+ + \lambda_i^f(t), \tag{1}$$

which is the evolution of the queue over time, where $[x]^+ \triangleq \max(x, 0)$, and $\lambda_f^i(t) \in A_f(t)$ represents the data arrival rate at node i, with $A_f(t)$ being the set of packets of flow f arriving at the source node, s_f. In this case we consider the queues to be operating in discrete time, $t \in \mathbb{Z}^+$, where $q_f^i(t)$ represents the queue length at node i. Point-to-point channel-power states for channel state, $h(t)$, and transmission power are used to realize the transmission rate as follows:

$$r_{i,j}^f(t) = B_{i,j}(t) \log_2 \left(1 + \gamma_{i,j}^f(t) \right), \tag{2}$$

where $B_{i,j}(t)$ represents the backhaul bandwidth, and $\gamma_{i,j}^f(t)$ is the SINR experienced by the traffic flow when transmitted via link (i, j), defined as follows:

$$\gamma_{i,j}^f(t) = \frac{p_{i,j}^f(t) g_{i,j}^f(t)}{\sum_{k \neq j}^K p_{k,j}(t) g_{k,j}(t) + N_0}, \tag{3}$$

where $p_{i,j}^f(t)$ represents the transmission power of transmitting flow f from IAB node i to j, $g_{i,j}^f(t)$ is the distance-dependent channel gain assumed to follow a Rayleigh fading distribution with unitary average power, $g_{i,j}^f(t) \sim \exp(1)$. The first term in the denominator is the aggregated interference from the access users, while the second term, N_0, is the white Gaussian noise spectral density. Since in IAB networks, part of the wireless spectrum is used for the backhaul connection of SBSs, the SBSs must be able to dynamically reserve resources for backhauling traffic to the gateway, n_0. That is, if $r_{i,j}^f(t)$ in (2) is the backhaul rate, then $r_{n,k}(t)$ represents the access rate. Therefore, based on this intuition, the access-backhaul condition can be stated as follows:

$$r_{i,j}^f(t) = B_{i,j}(t) \log_2 \left(1 + \gamma_{i,j}^f(t) \right) \leq r_{n,k}(t). \tag{4}$$

Using this condition, and also given the transmission power, the data rate for link (i, j) can be defined as $\sum_{f \in \mathcal{F}} r_{i,j}^f(t)$.

A. THE MARKOV DECISION PROCESS

Assuming that the proposed system follows a Markov process with discrete time steps, let t define the time intervals. The objective of the agent is to determine an optimal policy, π, that maps a state space, $\mathcal{S}$, onto an action space, $(\pi : \mathcal{S} \to \mathcal{A})$ that maximizes the expected reward R, while minimizing network delay [23]. Thus, an MDP is represented by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \mathcal{S}')$ [24], where $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \to [0, 1]$ is the unknown transition probability function, where $\mathcal{P}(s(t+1)|s, a)$ is the transition probability from state s to $s(t+1)$ after taking action a [25]. The current network observations constitute the state set, $\mathcal{S}$, i.e., available bandwidth, network load, i.e., the number of traffic flows and traffic demand, and network status, i.e., the channel conditions and interference levels. Generally, the state space can be summarized into utilization and the port rate as follows:

$$s_j(t) = \{U_{sw_i}(t), P_{z,sw_i}(t)\} \in \mathcal{S}, \tag{5}$$

where $U_{sw_i}(t) \in [0, 1]$ represents the current utilization of the flow table of switch i, and $P_{z,sw_i}(t)$ represents the port rate of port z of switch i. On the other hand, the RA decisions constitute the action set, which could be the spectrum and computational resources, as well as the network configurations. In this way, the action set, $\mathcal{A}$, consists of the route choice, power management, and the throughput, such that the i-th node scheduling action, $a_f(t)$, can be defined as the link to which the flow f is routed, and at the assigned transmission power, which can be defined as follows:

$$a(t) = d_j^f(t), \quad j \in \mathcal{V}, \ f \in \mathcal{F}. \tag{6}$$

Therefore, a scheduling policy π, which maps the system state, $s_f(t)$, to the scheduling action, $a_f(t)$, is defined such that $a_f(t) = \pi(s_f(t))$. The transition function of the MDP is denoted as $\mathcal{P}(j|i, l)$, which is the probability that $s_{i,f}(t+1) = j$ given that $s_{i,f}(t) = i$, and $a_f(t) = l$, is defined as follows:

$$\mathcal{P}(j|i, l) = \begin{cases} 1, & \text{if} \quad l = (j, z) \\ 0, & \text{otherwise.} \quad \forall j \in \mathcal{S} \end{cases} \tag{7}$$

This is the transition probability function, which must be the same for all packets in all traffic flows, such that the reward function can be represented as follows:

$$R_f^{\pi}(j) = \begin{cases} R(s, \pi(s)), & \text{if} \quad j = d_f \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

where $R(s, \pi(s))$ is the discounted reward for a packet in flow f for being in state j, defined as follows:

$$R(s, \pi(s)) = \mathbb{E}\left[\gamma^t \cdot r_{i,j}^f(t)\right], \tag{9}$$

where $0 \leq \gamma^t \leq 1$ is the discount factor. Then, the reward of taking action a under any state s can be defined using a reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S}' \rightarrow \mathbb{R}$. Therefore, the cumulative reward expectation of the access-backhaul condition in (4) is represented as follows:

$$R_f^{\pi}(j) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{T}\sum_{f\in\mathcal{F}}\sum_{(i,j)\in\mathcal{E}} R(s, \pi(s))\right], \tag{10}$$

where T denotes the horizon for which the system is observed.

B. FORMULATING THE CONSTRAINED MARKOV DECISION PROCESS

Since the action taken to maximize a certain reward always goes with an incurred cost, a cost function, $C_i : \mathcal{S}\times\mathcal{A}\times\mathcal{S}' \rightarrow \mathbb{R}$, is defined. Minimizing network delays requires that the transmission buffer has to be monitored for queuing delays and packet losses. In this way, a buffer cost is defined to reward the system for minimizing queuing delays, thereby protecting against overflows and subsequent packet losses, as well as penalizing every packet that is lost. The buffer cost is defined as the expected sum of holding costs and overflow cost with respect to the traffic arrival and goodput distributions [26], and can be expressed as follows:

$$g([q, p], \psi, y, z) = \sum_{\lambda=0}^{\infty}\sum_{f=0}^{z} p^{\lambda}(\lambda)p^{f}(f|\psi, z) \{[q-f] + \eta\max([q-f] + \lambda - Q, 0)\}, \tag{11}$$

where $[q - f]$ is the holding cost, which represents the number of packets that were in the buffer at the beginning of the time slot. Since a stable buffer is assumed, according to Little's theorem, the holding cost is proportional to the queuing delay [27]. The overflow cost, $\eta\max([q-f]+\lambda-Q,$ imposes the penalty η for each packet that is dropped.

$$C_f^{\pi}(j) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\sum_{f\in\mathcal{F}}\sum_{(i,j)\in\mathcal{E}} \gamma^t g(s, \pi(s))\right]. \tag{12}$$

The objective of this formulated constrained Markov decision process (CMDP) is to find a policy, π_θ, which maximizes (10), while satisfying (12).

C. THE OPTIMIZATION PROBLEM

We now formulate a stochastic optimization problem for maximizing the average sum throughput, subject to the capacity, queue stability, and power consumption constraints. Here, the optimization problem is expressed as follows:

$$P : \max_{p,\pi} R_f^{\pi}(j) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=1}^{T}\sum_{f\in\mathcal{F}}\sum_{(i,j)\in\mathcal{E}} R(s, \pi(s))\right], \tag{13}$$

$$\begin{aligned} &\text{subject to} \\ &C1 : C_f^{\pi}(j) \leq Q_{\text{th}}, \quad \forall j \in \mathcal{V} \\ &C2 : C_f^{\pi}(j) \leq C_e, \quad \forall (i,j) \in \mathcal{E}^+ \\ &C3 : p_{i,j}^f \geq 0, (i,j) \in \mathcal{V}| \sum_{j\in\mathcal{V}_i}\sum_{f\in\mathcal{F}} p_{i,j}^f \leq P_{i,\max}. \end{aligned} \tag{14}$$

The constraint **C1** ensures that buffer overflows and subsequent packet losses are prevented by forcing the queue length not to exceed the threshold Q_{th}. This constraint puts emphasis on the transmission delay by controlling the packet processing time per node, i.e., $0 \leq D_j(t) \leq \delta_j^f$, where $D_j(t)$ is the instantaneous delay of node j, and δ_j^f is the upper bound on the processing time. Based on the evidence in [28] that the node packet-processing capacity is a very important measure in minimizing delays when the λ_j^f is high, then $D_j(t)$ depends on the node processing capacity, i.e.,

$$Pr\{D_j(t) \geq d_{j,max}(t)\} \leq \delta_{th}, \tag{15}$$

where $d_{j,max}(t)$ is the maximum achievable delay of node j, while δ_{th} is the threshold of the probabilistic delay. The constraint **C2** ensures that the required backhaul capacity is always less than the capacity of the link C_e. This constraint means that the average backhaul transmission rate, $\bar{r}_{i,j}^f(t)$, has to be kept below the link capacity ensuring that the long-term arrival rate does not exceed the average transmission rate, which in turn prevents buffer overflows and subsequent packet losses, i.e., $\bar{r}_{i,j}^f(t) \geq \lim_{t\rightarrow\infty}\sum_i \frac{1}{T}\mathbb{E}\{\lambda_f(t)\}$. Lastly, the constraint **C3** emphasizes on the decision vector, p, π in (13), which defines the transmission power range contained in the transmission power vector, $\mathbf{p}$. This constraint ensures that the transmission power assigned to the forwarding node i does not exceed the maximum allowed transmission power by enforcing a power control condition.

IV. PROPOSED DEEP REINFORCEMENT LEARNING WITH RECURSIVE DISCRETE CHOICE MODEL

The proposed algorithm combines the DRL strategy with the RDCM to form a DRL-RDCM scheme that is suited for next generation routing applications since it can provide rapid and accurate route predictions. Here, the mechanism for adjusting the reward value is flexible, i.e., if choosing a route that is considered to be a bad route gives a low value

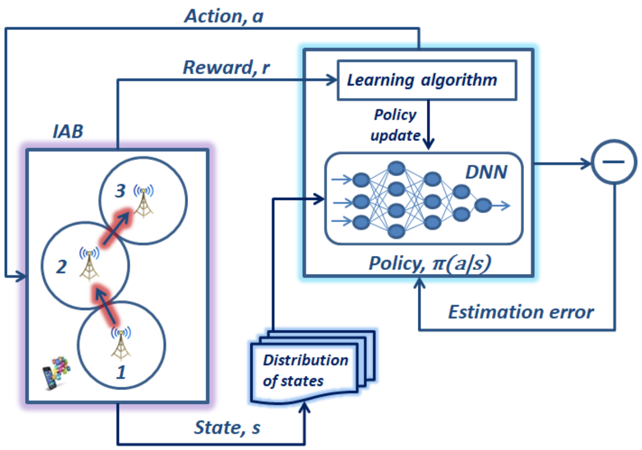


FIGURE 4. Deep reinforcement learning for modeling reward estimation in an IAB network.

of the punishment, the estimated value of that route slowly decreases and probably this "bad route" may be selected in the foreseeable future. On the other hand, if the value of the punishment is very high, the route may no longer be chosen in future routing events. The proposed framework tries to find a balance between low and high rewards/punishment using an efficient cost model. The proposed framework, which is based on a DRL strategy is illustrated in Fig. 4 below.

As shown in Fig. 4, the proposed DRL-RDCM learning algorithm computes and updates a policy, π, for the DNN agent to achieve a better level of performance and generality. The complexity of learning through trial-and-error is reduced using RDCM discussed in the following section. Evaluating the attributes of the IAB network such as throughput, average packet losses, and latency are the main targets of the routing problem. According to the routing nature and dynamics of the problem, a forward-backward exploration technique is used to learn the network attributes. Using this process, both the local and global network attributes update, where the long-term reward is related to the global network performance by looking for routes with the highest success rate. With reference to the graph model in Fig. 3 above, the source node conveys route setup request messages to the destination, i.e., the gateway/MBS. The reward generation and value estimates are presented on a link-by-link basis, and the rewards are propagated based on acknowledgement messaging from other network nodes. In this case, the local reward value is directly related to the receipt of the acknowledgement message for a packet successfully received by the j-th node. A higher value of the local reward means that node j is a good candidate for a link towards the gateway, which increases the probability of being selected for backhaul route establishment in future. Therefore, the set of nodes that are adjacent to node i in this probabilistic graph are referred to as its physical neighbors, and finding the best route between the source and the destination must be rewarded. Since the reward function is intimately tied to the state and action spaces, the Q-learning algorithm is used. In a time-dependent problem such as this one, the distance to the reward is handled using the DRL strategy, where the agent is trained to interact with the environment, and the goal is to maximize the total rewards and to also learn by adjusting its strategy based on the rewards.

A. THE RECURSIVE DISCRETE CHOICE MODEL

According to the IEEE 802.16 standard, a logical link between two nodes in a network can be set up provided they are able to communicate directly with each other [35]. The route choice model proposed in this work is based on the assumption that nodes behave rationally by maximizing a certain utility function, or equivalently minimizing a certain cost function [29]. In addition, the nodes observe additional parameters that affect their path choice. These factors vary across nodes and they are unknown to the model. As such, a random term, ϵ, is added to the cost function. Although the modeller would not know the additional parameters, it knows the family of distributions for ϵ. The objective then is to infer the probability that a given path is optimal given the current distribution of states. In this work, a recursive discrete choice model is incorporated in the DRL strategy, with the objective of inducing a Markov chain into the graph $\mathcal{G}$. Considering the graph model in Fig. 2, the sets of edges entering and leaving node j are denoted as $\mathcal{E}_j^-$ and $\mathcal{E}_j^+$, respectively. The links or paths passing through node j are represented by e_j, and the route choice model is developed using discrete choice experiment, which incorporates choice overhead by means of a penalty parameter [30]. Choice aversion is computed for each outgoing edge, $e_j \in \mathcal{E}_j^+$, and a collection of i.i.d. random variables of the error terms, $\{\epsilon_{e_j}\}_{e_j \in \mathcal{E}}$, are assumed such that the RDCM becomes a recursive logit model [31]. The recursive reward/utility associated with edge e_j is then defined as follows:

$$R_f^{\pi}(e_j) = R_{e_j}^f + \mathbb{E}\left(\max_{e' \in \mathcal{E}_j^+} \{V_{e'} - \Omega_{j_e} \log |\mathcal{E}_j^+|\}\right), \quad (16)$$

where $R_{e_j}^f$ is the instantaneous reward of edge e_j and the expectation $\mathbb{E}(\cdot)$ is the adjusted continuation value associated with regard to the choice of edge e_j, V_e is the observed realization of random rewards. The factor $\Omega_{j_e} \log |\mathcal{E}_i^+|$ represents the penalty that captures the size of the choice set, i.e., $\mathcal{E}_{je}^+$, where the parameter $\Omega_{j_e} \geq 0$ is the parameter representing choice aversion [31]. Thus, assuming that the collection of random variables at each node $j \neq n_0$ fulfills the sufficient criterion whose distribution is sufficiently scaled as defined in [32], (16) can be reformulated as follows:

$$R_f^{\pi}(e_j) = R_{e_j}^f + \log\left(\sum_{e' \in \mathcal{E}_{je}^+} e^{V_{e'}}\right) - \Omega_{j_e} \log |\mathcal{E}_{je}^+|, \quad (17)$$

where the second and third terms represent the closed-form expression of the expectation in (16). Since each flow has to find an optimal route to n_0, when the flows reach node

$j \neq n_0$ they observe the realization of random utilities, $V_e, \forall e_j \in \mathcal{E}_j^+$, and subsequently choose the edge with the highest utility. This is done by leveraging regret learning, which exploits information about channel states, h, and queue states, Q, in choosing the optimal route [33]. This intuition is influenced by the learning framework in Fig. 3, where the forward and backward exploration are employed in learning maximization of the long-term utility of traffic flows. This whole process is repeated at each subsequent node, $j' : j \neq n_0$, resulting in a RDCM. Therefore, the expected traffic flow entering a node will take an outgoing route according to a choice probability defined as follows:

$$\mathcal{P}(e_j|\mathcal{E}_j^+) = \mathcal{P}\left(e_j = \arg\max_{e'\in\mathcal{E}_j^+} V_{e'}\right). \quad \forall j \neq n_0 \qquad (18)$$

It must be noted that as the value of the parameter Ω_{j_e} increases, the edge choice probability (18) is increasingly penalized by the size of the choice set. This reflects the cost of choice overload onto the edge utility of the user with a large choice set. According to the law of flow conservation [34], $x_j = \sum_{e\in\mathcal{E}_j^-} f_e, \forall j \neq n_0$ is feasible if there exist a unique flow vector that satisfies all flow constraints. Therefore, the solution of this RDCM can be equivalently written in the form of route choice probabilities, assuming that for each route the utility associated to it is a random variable defined as follows:

$$\begin{aligned}\tilde{R}_f(e_j) &= \sum_{e\in\mathcal{E}}(R^f_{i,j} - \Omega_{j_a}\log|\mathcal{E}^+_{j_a}|)\\ &= \sum_{e\in\mathcal{E}} R^f_{i,j} - \sum_{e\in\mathcal{E}} \Omega_{j_a}\log|\mathcal{E}^+_{j_a}|. \qquad (19)\end{aligned}$$

Therefore, under these conditions, using the choice probability in (18), the probability of choosing the route e_j can be defined as follows:

$$\mathcal{P}_e \triangleq \mathcal{P}\left(e_j = \arg\max_{e'\in\mathcal{E}} \tilde{R}_f(e_j)\right), \qquad (20)$$

which is equivalent to the greedy action selection in [46] equation (34). Among the possibly multiple routes that the flows can take between source and destination, the algorithm selects only one. Flow regulation of rate and delay at ingress can only be ensured along a single path, hence resource utilization bounds need to be established.

B. FORMULATION OF OPTIMIZATION BOUNDS

The existence of the non-linear probabilistic constraint (15) in **C1** makes the optimization problem difficult to solve. In order to circumvent this issue, its linear deterministic equivalent is introduced using Markov's inequality such that for a non-negative random variable X and $a > 0$, one can have $Pr\{X \geq a\} \leq \mathbb{E}[X]/a$, which results in the following [36]:

$$Pr\left\{\frac{q_i^f(t)}{\lambda_f} \geq \delta_{th}\right\} \leq \frac{\mathbb{E}[q_i^f(t)]}{\lambda_f \delta_{th}}. \qquad (21)$$

The mathematical models of queues with deadlines and rewards are used to describe the attributes of the system, such that if the utilization factor follows the accurate stability conditions described in [37], the stability of the system can be guaranteed. Therefore, in order to relax (15), the condition of the expected queue length must be satisfied as follows:

$$\mathbb{E}[Q_f^i(t)] \leq \lambda_f \delta_{th} \delta_j^f, \quad \forall f \in \mathcal{F}, \ \forall t \in T. \qquad (22)$$

In order to guarantee that all flows have a certain minimum level of QoS, a minimum requirement $r_{i,j}^{min}$ is introduced as follows:

$$r_{i,j}^{min}(t) \leq r_{i,j}^f(t) \leq r_{i,j}^{max}(t), \qquad (23)$$

where $r_{i,j}^{max}$ is the maximum rate constraint, which is enforced to avoid the over-allocation of resources when a large number of packets are sent simultaneously, such that $r_{i,j}^f(t) \gg q_f^i(t)$. Then, the optimization problem can be rewritten as follows:

$$P^* : \max_{\bar{r},\pi} \sum_{t=1}^{T} \sum_{f\in\mathcal{F}} \sum_{(i,j)\in\mathcal{E}} \omega_f R_f^{\pi}(t), \quad \text{s.t.} \quad (23), \qquad (24)$$

where ω_f is a weight assigned to each flow f. Then, the expected queue length can be defined as follows:

$$\mathbb{E}[Q_f^j(t)] = t\lambda_f^j - \sum_{\tau=0}^{t} r_f(\tau). \qquad (25)$$

By substituting (25) into (22), the minimum rate requirement can be obtained as follows:

$$r_{i,j}^f(t) \geq \bar{\lambda}_f(t - \beta_f \delta_f) - \sum_{\tau=1}^{t-1} r_{i,j}^f(\tau). \qquad (26)$$

Since the statistical information regarding all the candidate routes are not available, a proper solution to (13) is difficult to obtain. Then, using the reward function in (8) and the effective throughput in (9), the CMDP equivalent of (10) can be represented as follows:

$$\max_{\pi} \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \sum_{f\in\mathcal{F}} \sum_{(i,j)\in\mathcal{S}} \sum_{\tau=0}^{\tau_f} R_f(s_{i,f}^{\pi}(t+\tau))\right], \qquad (27)$$

subject to

$$\lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \sum_{f\in\mathcal{F}} \sum_{(i,j)\in\mathcal{S}} \sum_{\tau=0}^{\tau_f} \mathbb{I}\{a_{i,f}^{\pi}(t+\tau) = e\} \leq C_e\right], \qquad (28)$$

where the $\mathbb{E}[\cdot]$ is the expectation taken with respect to the traffic flow arrival process, the transition function, and the optimal policy π. At this point, the policy, π, continues to generate the system states. In order to solve the formulated CMDP in (24), the Lagrange duality equivalent of the problem is formulated, where it is assumed that the problem is associated with a Lagrangian, $\mathcal{L}$. The Lagrangian equivalent of (27) and (28) can be written as follows:

$$\mathcal{L}(\pi, \kappa) = \sum_{e\in\mathcal{E}} \kappa_e C_e + \lim_{T\to\infty} \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T} \sum_{f\in\mathcal{F}} \sum_{(i,j)\in\mathcal{E}} \sum_{\tau=0}^{\tau_f}\right.$$

$$\left.\left(R_f(s_f^{\pi}(t+\tau)) - \sum_{e_j\in\mathcal{E}} \kappa_e \mathbb{I}\{a_f^{\pi}(t+\tau)=e\}\right)\right], \tag{29}$$

where $\kappa > 0$ is the Lagrange multiplier. Then, for every feasible policy, $\pi \in \Pi$, it can be observed that (27) is bounded below by the formulated $\mathcal{L}(\pi, \kappa)$. Therefore, if the rewards and transition probabilities are the same for every packet in a given traffic flow are the same, then the state-value function can be defined as follows:

$$V_f^{\pi}(\kappa) = \mathbb{E}\left[\sum_{\tau=0}^{\tau_f}\left(R_f(s_f^{\pi}(t+\tau)) - \sum_{e}\kappa_e\mathbb{I}\{a_f^{\pi}(t+\tau)=e\}\right)\right]. \tag{30}$$

where $\mathbb{E}[\cdot]$ is the expectation with respect to the underlying transition probability under the policy $\pi_f(\kappa)$. The Lagrangian in (29) can be written as follows:

$$\begin{aligned}\mathcal{L}(\pi,\kappa) &= \sum_{e\in\mathcal{E}}\kappa_e C_e + \sum_{f\in\mathcal{F}}\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{(i,j)\in\mathcal{V}} V_f^{\pi}(\kappa)\\ &= \sum_{e\in\mathcal{E}}\kappa_e C_e + \sum_{f\in\mathcal{F}}\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}|\mathcal{E}|V_f^{\pi}(\kappa)\\ &= \sum_{e\in\mathcal{E}}\kappa_e C_e + \sum_{f\in\mathcal{F}}\rho_f V_f^{\pi}(\kappa).\end{aligned} \tag{31}$$

Then, the dual function is obtained as follows:

$$D(\kappa) = \max_{\pi}\mathcal{L}(\pi,\kappa), \tag{32}$$

and the dual policy is represented as follows:

$$\pi(\kappa) = \arg\max_{\pi}\mathcal{L}(\pi,\kappa), \tag{33}$$

and the optimal dual variable is denoted as follows:

$$d^* = \arg\min_{\kappa\geq 0} D(\kappa). \tag{34}$$

Assuming that there is no duality gap, the optimal policy, π^*, of the CMDP is the same as $\pi(d*)$, and that κ and $V_f^{\pi}(\kappa)$ of any flow are independent of all the other flows. In this case, the objective is to obtain the optimal policy, $\pi_f(\kappa)$, for each flow as follows:

$$\begin{aligned}D(\kappa) &= \max_{\pi}\mathcal{L}(\pi,\kappa)\\ &= \sum_{e\in\mathcal{E}}\kappa_e C_e + \max_{\pi}\sum_{f\in\mathcal{F}}\rho_f V_f^{\pi}(\kappa)\\ &= \sum_{e\in\mathcal{E}}\kappa_e C_e + \sum_{f\in\mathcal{F}}\rho_f\max_{\pi_f} V_f^{\pi_f}(\kappa)\\ &= \sum_{e\in\mathcal{E}}\kappa_e C_e + \sum_{f\in\mathcal{F}}\rho_f V_f^{*}(\kappa),\end{aligned} \tag{35}$$

where $V_f^*(\kappa) = \max_{\pi_f} V_f^{\pi_f}(\kappa)$, and $\pi_f(\kappa) = \arg\max_{\pi_f} V_f^{\pi_f}(\kappa)$. At this point, $\pi_f(\kappa)$ and $V_f^*(\kappa)$ can be computed using finite horizon dynamic programming.

C. COMPUTING THE COST FUNCTION

The ability of the novel cost function to maintain prior knowledge enables it to accelerate the convergence of the algorithm, which improves the algorithm's performance in delay-power trade-off. Since the transition probabilities as well as the link probabilities cannot be known a priori, a post-decision state-based dynamic programming technique is employed to compute the cost function. Here, the transition probability function is split between the known and the unknown dynamics in order to learn the link probabilities and obtain the optimal policy. Under the assumptions on arrival and processing rates, the analytical results for the buffer occupancy status are used to compute the cost function. The queue load and the delay distribution are taken as the known information and are exploited to develop a more efficient cost function based on the CMDP. In this case, a post-decision state, $\tilde{s}$, is defined, which is related to the current state as follows:

$$\begin{aligned}\tilde{s}(t) &= \left(\tilde{q}_j(t), h(t), x(t+1)\right)\\ &= \left([q_j(t)-\mu_j(t)], h(t), x(t+1)\right).\end{aligned} \tag{36}$$

The post-decision state in (36) represents the state of the transmission buffer after packets have been transmitted, but just before new packets arrive, thus, the queue length can be represented by $\tilde{q}(t) = q_j(t) - \mu(t)$. Here, the channel state is assumed to be the same as the state at time t, and the power management post-decision state is the same as the power management at time $t+1$. Then, the state at time $t+1$ can be represented as follows:

$$\begin{aligned}s(t+1) &= (q(t+1), h(t+1), x(t+1))\\ &= ([q(t)-\mu(t)]+\lambda(t), h(t+1), x(t+1)).\end{aligned} \tag{37}$$

It must be noted that at the state $s(t+1)$, unknown dynamics, such as the arrival rate and the channel state, have been incorporated. The post-decision queue state after traffic arrival can be represented as $q(t+1) = \tilde{q}(t) + \lambda_f^j$. The introduction of the post-decision state enables the factorization of the transition probability function into known and unknown components. In this case, the known component accounts for the transition from the current state, $s(t)$, to the post-decision state, $\tilde{s}$. On the other hand, the unknown component accounts for the transition from the post-decision state, $\tilde{s}$, to the next state, $s(t+1)$. Factorizing the transition probability function results in

$$p(s(t+1)|s,a) = \sum_{s} p_u(s(t+1)|\tilde{s},a)p_k(\tilde{s}|s,a), \tag{38}$$

where subscript k represents the known component and subscript u represents the unknown component. Since the queue overflow depends on the arrival distribution, which is an unknown component, the queue overflow cost may depend on the action and the post-decision state. Based on the goodput distribution in (11), the cost function can similarly be factorized as follows:

$$c(s,a) = c_k(s,a) + \sum_{\tilde{s}} p_k(\tilde{s}|s,a)c_u(\tilde{s},a). \tag{39}$$

Since the goodput distribution has to account for packet losses, the algorithm must penalize packet overflows. Since action exploration is not necessary to learn the optimal policy, the known transition probability function can be defined as follows:

$$p_k(\tilde{s}|s,a) = p^x(\tilde{x}|x,y)p^f(q-\tilde{q}|\psi,z)I(\tilde{h}=h), \tag{40}$$

and the unknown transition probability can be represented as follows:

$$p_u(s(t+1)|\tilde{s}) = p^h(h(t+1)|\tilde{h})p^\lambda(q(t+1)-\tilde{q})I(x(t+1)=\tilde{x}), \tag{41}$$

where $I(\cdot)$ is the indicator function, which takes a value of 1 if its argument is true, and 0 otherwise. Then, the known and unknown cost functions are defined as follows:

$$c_k(s,a) = \rho([h,x],\psi,y,z) + \mu\sum_{\mu=0}^{z}(\mu,\psi,z)[q-\mu] \tag{42}$$

and

$$c_u(\tilde{s}) = \mu\eta\sum_{\lambda=0}^{\infty}p^\lambda(\lambda)\max(\tilde{q}+\lambda-\mathcal{Q},0), \tag{43}$$

where the parameter η represents the penalty. The post-decision value function, $\tilde{V}^*$, which plays a similar role as the action-value function in Q-learning can be used to represent the unknown component of the discounted cost as follows:

$$\tilde{V}^*(\tilde{s}) = c_u(\tilde{s}) + \gamma^t\sum_{s(t+1)}p_u(s(t+1)|\tilde{s})V^*(s(t+1)). \tag{44}$$

The minimization of the cost function can be obtained by substituting the unknown component into the known one as follows:

$$V^*(s) = \min_{a\in\mathcal{A}}\left\{c_k(s,a) + \sum_{\tilde{s}}p_k(\tilde{s}|s,a)\tilde{V}^*(\tilde{s})\right\}. \tag{45}$$

Then, the optimal policy of the post-decision state-value function can be computed as follows:

$$\pi^*_{post}(s) = \min_{a\in\mathcal{A}}\left\{c_k(s,a) + \sum_{\tilde{s}}p_k(\tilde{s}|s,a)\tilde{V}^*(\tilde{s})\right\}. \tag{46}$$

In order to keep the system at equilibrium, when the queue length approaches its maximum, the system has to quickly generate a policy for an optimal action to reduce the queue length by increasing the transmission rate $r(t)$. As such, QoS parameters such as packet goodput and packet holding costs are considered to account for the increase in transmission power as the transmission rate increases.

V. ALGORITHM DESCRIPTIONS AND COMPUTATIONAL COMPLEXITIES

In this section, the basic formulation of the proposed DRL approach, which uses a DNN, is introduced together with the associated computational complexities for the proposed algorithm and other baseline algorithms used for comparison. The training and inference phases of the proposed algorithm are separated in order to improve clarity and understanding of the analysis of the computational complexity.

A. DNN OPTIMIZATION AND ACTION SELECTION

A feedforward multi-layer perceptron (MLP) neural netrowk (NN)is used for the training process and action output receives input data for routing in the IAB network. Information about a known network such as the topology and link capacities are required when training the DNN. This includes a known source, destination, bandwidth, duration, as well as time of arrival in order to obtain the temporal sequence of traffic flows. The topology of the DNN that is implemented by the agent in the DRL strategy is a feedforward MLP NN with linear hidden neurons and sigmoid output neurons [38]. Another important requirement is the time series, i.e., the knowledge of the traffic passing over the network in a certain period of time. In this case, a dataset with topology and aggregated information about traffic, which comes in the form of an $N \times N$ traffic matrix was used, where the element in row i and column j represents the total amount of traffic, i.e., the average bandwidth in a certain period of time between nodes i and j. With the state space shown in (5), the optimization of the MLP was done using the analysis of the number of neurons in the hidden layer, and using three training sets, i.e., training, validation, and testing. The input data consists of the node ID of the packet that should be transferred through to the gateway. The interface status or utilization represents the information about the status of all interfaces for the node/router.

B. SAMPLING OF ROUTE CHOICE PROBABILITIES

1) MAX-WEIGHT AND BACK-PRESSURE ALGORITHMS

Since some nodes may fail due to power issues, damage, congestion, as well as environmental interference, this should not affect the overall task of the IAB network. In order to avoid such catastrophes, the routing protocol must be able to use the information at its disposal and find alternate routes toward the gateway. The max-weight and back-pressure algorithms were used to determine which link(s) should be activated. The max-weight checks the maximum queue at any node and it gives out the results accordingly, while the back-pressure algorithm compares two nodes to determine which link should be activated. In order to obtain better results, no restrictions must be put on the weights of the DNN. This is because of the existence of more than one possible route, whereby a choice of the best route creates a stochastic decision-making problem. Situations like this cannot be handled using decision trees due to their instability when slight changes are introduced, which is prevalent in wireless networks. Therefore, a cumulative prospect theory is applied to the Metropolis-Hastings algorithm [41]. In this way, a POMDP with Q-values that quantify the agents' value of choosing one route over another is implemented. The agent then calculates its prediction error, which is equivalent to reward minus the Q-value of the decision. The prediction error, the belief state, the learning rate, and the reward are then used to update the Q-values of the next iteration. The procedure for training the DNN is outlined in **Algorithm 1** below:

Algorithm 1 Procedure for Training the Deep Neural Network

```
Input: State, $s \in \mathcal{S}$,; Learning rate, $\alpha_t \in \mathcal{A}$;
Input: Discount factor, $\gamma^t$
Output: $Q(s, a)$
Initialize environment for IAB network;
For each state, $s \in \mathcal{S}$, do
    Randomly pick $w_1, \cdots, w_d$ according to $\mathcal{N}(\mu(s), \sigma_s)$
    For each iteration of the training episode do
        Find step length and sample minibatch of input data, and
        Run SGD and update weights
    End For
    Determine available action $a \in \mathcal{A}$ and estimate $Q(s, a)$
End For
Return $Q(s, a); \theta)$
```

2) SAMPLING OF WEIGHTS

The value of the perceived weight is randomly sampled from a normal distribution $\mathcal{N}(\mu(s), \sigma_s)$ as shown in **Step 03** of **Algorithm 1**. This is accomplished by using the Metropolis-Hastings algorithm, where $\mu(s)$ is the mean and σ_s is the standard deviation. The value of the standard deviation is one of the parameters of the proposed model. Each sampling event results in a stimulus to which noise it added to, then the agent creates a belief state that determines the correctness of the stimulus. This belief state is then applied to model the behavior of IAB nodes in the context of route choices by demonstrating the validity of the discrete choice model in route choices. From each belief state, available actions are determined to estimate the Q-value, $Q_{j,s}$. The application of cumulative prospect theory models the effect of the learning rate and noise levels on the cumulative reward, $r_{i,j}(t)$. Therefore, the Q-value of making a state transition can be represented as follows:

$$Q_{i,j}(t) = r_{i,j}(t) + \gamma^t \max_s Q_{j,s}, \tag{47}$$

where γ^t is the discount factor. The agent then combines the formed belief state as to the current side of the stimulus with its stored Q-values, then chooses a particular route and receives an appropriate reward, $r_{i,j}(t)$. In this way, the value of $r_{i,t}(t)$ becomes the second parameter of the proposed model. The higher the Q-value, $Q_{i,j}(t)$, the higher is the probability that the agent will choose that particular route over the other alternative routes.

C. ACTION SELECTION AND REWARD COMPUTATION

The process of reward value adjustment should be flexible, i.e., the adjustment may not be too small to not cause changes or too large to induce sudden change due to a specific event. As aforementioned, if the value of the punishment for choosing a bad route is very small, the estimated value of that route will slowly decrease and probably this bad route can still be chosen for a long time. On the other hand, if the punishment value is too high, a route may no longer be chosen because of just one packet loss event. Thus, a balance should be struck between very low and very high rewards/punishment. Therefore, the objective of this section is to evaluate the reliability of backhaul routes in terms of the cost of delays and power. The difference between the conventional Q-learning algorithm and post-decision state learning is that, instead of using a sample average of the action-value function to approximate Q^*, the latter uses a sample average of the post-decision value to approximate $\tilde{V}*$. In the post-decision state learning algorithm, the state space is characterized by the buffer state, and the only action is the throughput, subject to packet losses. As the algorithm updates the state-action pair, it only provides information about the buffer-throughput pair. The post-decision state learning provides information about every state-action pair that can potentially lead to all the corresponding buffer-throughput pairs. It is worth noting that here, the experience tuples are updated in parallel, as such the post-decision state learning algorithm has the same memory requirements as the DRL algorithm. The procedure for the proposed DRL strategy is outlined in **Algorithm 2** below:

D. COMPUTATIONAL COMPLEXITY OF THE PROPOSED ALGORITHM

The whole operation of training the DNN and action selection has a run-time complexity of $\mathcal{O}(n)$ in forward propagation as well as in the backward propagation. Then, the run-time computational complexity of both the forward and backward propagation can be obtained as $\mathcal{O}(n \cdot n) = \mathcal{O}(n^2)$ [40]. The outputs of the DNN agent are the actions that also serve as the input to the Q-learning algorithm, whose first task is to select the action that maximizes the reward. Since the evolution metric used is the total reward collected by the agent in every training episode, the configuration of the learning rate is quite critical. The DRL considers obtaining the long-term reward by performing the choice evaluation/aversion procedure to generate higher rewards and lower costs. This evaluation process is carried out using the defined graph structure and the RDCM, where a number of candidate routes are evaluated in terms of utility and cost. Therefore, the learning update

Algorithm 2 Procedure for DRL With RDCM

```
Input: $\lambda_j^f(t)$; Buffer size, $\mathcal{Q}$; $T$, $\alpha_t$, $\gamma^t$
Output: Reward, $\pi^*(q(t))$, $\hat{r}(t)$, $c(q(t), y(t))$
Initialize buffer occupancy as $q(t)$
Initialize post-decision state value function $\tilde{V}^0$
Create candidate set of routes for traffic flow
For each link $(i, j)$ do
    Find link to nearest node and observe SINR
    If current SINR $\gamma_{i,j} \geq \gamma_0$ then
        Select $a_f(t) = \arg\max_a Q(s_f(t), a_f(t); \theta)$
        Take transmission action, i.e., $a_f(t) = \arg\min_{a \in \mathcal{A}} \left\{ c_k(s, a) + \sum_{\tilde{s}} p_k\left(\tilde{s}|s, a\right) \tilde{V}(\tilde{s}) \right\}$
        Observe transition to next state, $s(t+1)$
    Else
        Request route on another candidate link
    End If
    Observe the post-decision state experience tuple $\tilde{\sigma} = (s(t), a(t), \tilde{s}, c_u, s(t+1))$
    Populate transition probabilities, $(s(t), a(t), p(t), r(t), s(t+1))$
End For
Return $\pi^*(q(t))$, $\hat{r}(t)$, $c(q(t), y(t))$
```

and the computation of the reward and cost result in run-time complexity of $\mathcal{O}(n^2)$. The computation of the cost function through the use of the value iteration approach has a sample complexity of $\mathcal{O}(n^2)$, which is similar to the results found in [42]. On overall, the post-decision state learning algorithm does not require more memory than the Q-learning algorithm used in the RL strategy, and therefore the computational complexity of the post-decision state learning algorithm can be determined as $\mathcal{O}(n \cdot n^2) = \mathcal{O}(n^3)$.

E. DESCRIPTION OF BASELINE ALGORITHMS

Due to the limitations surrounding the proper utilization of resources in IAB networks as well as the nature of the route requests and discoveries, a reliable benchmark algorithm has not been identified. To this effect, based on the stochastic nature of IAB networks, the traditional DRL and the generative model-based learning (GMBL) approaches were selected as benchmark algorithms for this work.

1) GENERATIVE MODEL-BASED LEARNING

The GMBL is a naive "plug-in" model-based ML technique used to build maximum likelihood estimates of the transition model in the MDP from observations and then find an optimal policy. It operates by preserving a local linear relationship utilizing the Laplacian matrix with the aim of maintaining the graph-based structure of the original data in Hamming space [43]. This technique follows a procedure under which each link is sampled a predefined number of times in order to determine its statistics to a desired level of accuracy. Then, the resulting model is used as an input to the CMDP framework. Moreover, by automatically assigning weights for each view to improve clustering performance, the method takes distinctive contributions of multiple views into consideration. In this work, an alternating iterative optimization method is designed to solve the resulting optimization problems. It is, however, difficult to implement since all nodes have to generate packets on their own in order to sample links. The sample complexity of the GMBL is proportional to the number of links in the network topology - consistent with the number of unknown parameters such as link success probabilities. In terms of sample and computational complexity of obtaining the ϵ-optimal policy in this model, the agent accesses the underlying transition model via a sampling oracle that provides a sample of the next state when given any state-action pair as input [44].

2) DEEP REINFORCEMENT LEARNING

DRL is a sub-field of ML that combines RL and deep learning (DL), where RL considers the problem of a computational agent learning to make decisions by trial and error. By incorporating DL into the solution, DRL allows agents to make decisions from unstructured input data without manual engineering of the state space. As a result, the equilibria of this strategy differs along three task complexity measures, i.e., (i) the cardinality of the choice space, where a state is equivalent to the information set facing the player along the path leading to the equilibrium; (ii) the level of iterative knowledge of rationality, and (iii) the level iterative knowledge of the strategy [45]. The greedy action selection of game theory and RL is illustrated with an almost similar complexity. However, more information is integrated in the RL strategy with the learning update, and as more information is integrated into an algorithm, it becomes more computationally complex to implement. The state-of-the-art RL strategy in resource allocation states that the computation time cannot be upper bounded by less than $\mathcal{O}(n^3)$.

VI. PERFORMANCE EVALUATION

A. NETWORK MODEL SETUP

In the experimental setup, the IAB network was deployed according to the standards of 5G standalone deployment, with the 5G core network completely disconnected from the 4G EPC. The radius of the deployment area is 1000 meters, and the donor node is 250 meters from the nearest IAB node, while the IAB nodes are 100 meters apart. A random walk model was adopted in simulating UE mobilities. Since wireless connections cannot be connected to the server at the same time, each user activates its wireless connection to the server using 802.11 links. To avoid collision and to provide better QoS to traffic flows in the dynamic network, time-sharing is employed. An initial randomized policy was set to a uniform distribution, and the inter-arrival time of a Poisson arrival process is an exponential random variable. In this way, the local reward is given to the route that has the best rate of success in delivering packets within their deadlines. The

TABLE 2. Simulation parameters.

Parameter	Value	Unit
Component carrier frequency	28	GHz
System bandwidth, B	20	MHz
Subcarrier spacing	60	kHz [48]
Maximum transmission power	80	mW
Maximum number of nodes, N	5	-
Finite buffer size, Q	25	packets
Fixed symbol rate, $1/T_s$	500 $\times 10^3$	symbols/sec
Time slot duration, Δt	0.5	ms
Finite horizon, T	125	time slots
Packet arrival rate	10	packets/slot
Base station processing time	0.6	msec/request
SDN controller processing time	0.2	msec/request
Discounting factor, γ^t	0.75	-

NetworkX library in python for producing random graphs from a given set of edges [47] was used to set up the network.

B. SIMULATION PARAMETERS

The main simulation parameters adopted from [46] are tabulated in TABLE 2 below.

To evaluate the performance of the proposed strategy, the assumption of network heterogeneity was made. The simulations, which affirm the potential of the proposed algorithm, were conducted using MATLAB^R R2021b software running on a workstation computer with an i5 Intel Core processor and a 3.2 GHz processor speed. Here, 100 SBSs were deployed in a randomly distributed manner over a 1000 radius, with the gateway placed towards the end of the network as shown in Fig. 1. Small cell connection distances that are indicated in Fig. 2, were set to unity, which means that the route lengths are measured in number of hops and delays are measured using queue lengths and waiting times.

C. DNN TRAINING PERFORMANCE

The training, validation, and testing sets were created based on the topology of the IAB network, and the number of samples for training the model depends on the kind of router for which the DNN agent was created. For each DNN layer, a matrix multiplication and an activation function are computed in forward propagation, and the rectified linear unit (ReLU) in the hidden layers computes the transfer function [39]. Without putting any restrictions on the weights, each threshold-activated neuron was simulated with a sigmoid activation at the output by computing the transfer function. Using the online approach, testing of the online training framework is done where the DNN is continuously being trained while being applied to the IAB network. The ability of the proposed approach to adopt accuracy in IAB routing, as well as the ability of the model to continuously learn from ongoing interactions with the IAB network and automatically re-adapt on the fly to changing dynamics is evaluated. The performance evaluation is done for 40 epochs, which each epoch run over 50,000 iterations, and the training results are shown below:

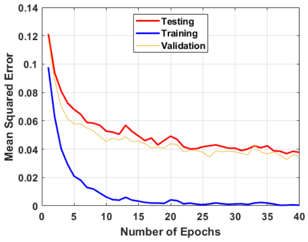


FIGURE 5. Training, testing, and validation loss using the mean squared error.

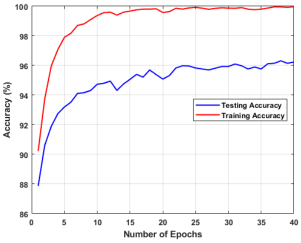


FIGURE 6. Training and testing accuracy.

Figure 5 above shows the test errors of the SGD as a function of the number of epochs. This is shown using the average MSE per training epoch, and it indicates the good performance of the algorithm in all the three aspects, i.e., training, testing, and validation. In the implementation of the SGD, the speed of convergence was enhanced by initializing the weights using heuristics, and by using Nesterov's momentum [49] and dropout. It is also apparent that the algorithm performs well as the distance between the testing and validation curves is minimal. Fig. 6 shows the average accuracy as a function of the number of epochs, i.e., training and testing per epoch. These results indicate a good performance of the algorithm in terms of learning from the data set, as the training accuracy immediately peaks at $\geq$ 90%, while the testing accuracy reaches 90% accuracy after two epochs. Both these plots are a little noisy, giving the impression that the training algorithm is not making steady progress. However, there is an indication that good results would be obtained when the real network data is used to train the system. On overall, these results indicate that the performance loss in terms of training and testing is already low after five training epochs, which suggests that the MLP can be adopted for the IAB problem under consideration.

D. RESOURCE AVAILABILITY AND RESOURCE DEMAND

In this subsection, the performance of scheduling and backhaul route selection is evaluated using route prediction probabilities to improve the understanding of the relationship between two variables against each other from a traffic trace obtained from a 5G standalone testbed at the National Chiao Tung University, China. The results in this subsection show how two dependent variables on two different axes vary with a

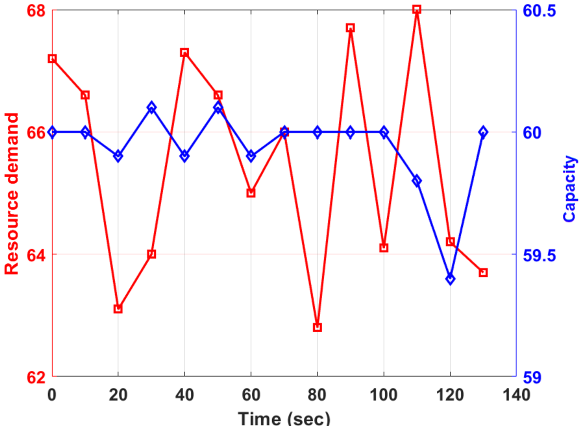


FIGURE 7. Resource demand and capacity vs time.

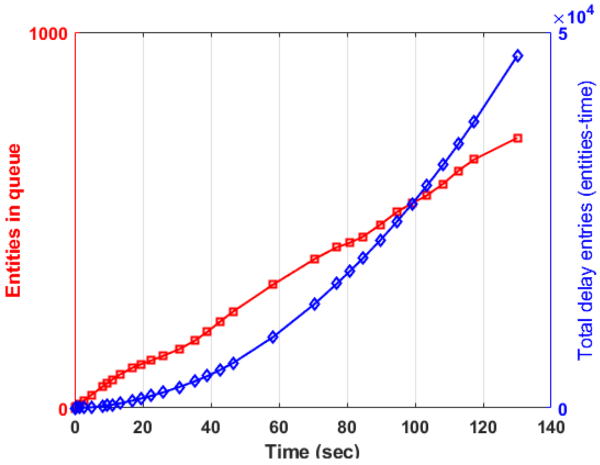


FIGURE 8. Variation of entities in a queue and total delay entities with time.

common independent variable. The time-series evaluation of the resource demand and capacity is shown in Fig. 7 below:

In Fig. 7 above, it can be seen that in most instances the resource demand is more than the available capacity, which means that the packet arrival rate is more than the departure rate, thus putting the system under pressure. The instability of the system is shown by the lack of proper correlation between the resource demand and the capacity. The time-series evaluation of the number of entities in the node queue and the total delay is shown in Fig. 8 below:

As shown in Fig. 8 above, the number of entities in the queue increases almost linearly with time, while the total delay increases exponentially with time. This behavior is expected since the Poisson arrival rate is modeled as an exponential function. This Poisson-exponential behavior adequately describes the first order auto-regressive model in a manner that is independent of the average level of the queue. When considered from a classical statistics point of view, each observation would be assumed as a sample of a given random variable, which would usually be i.i.d., thus neglecting the correlation of the evolutionary process.

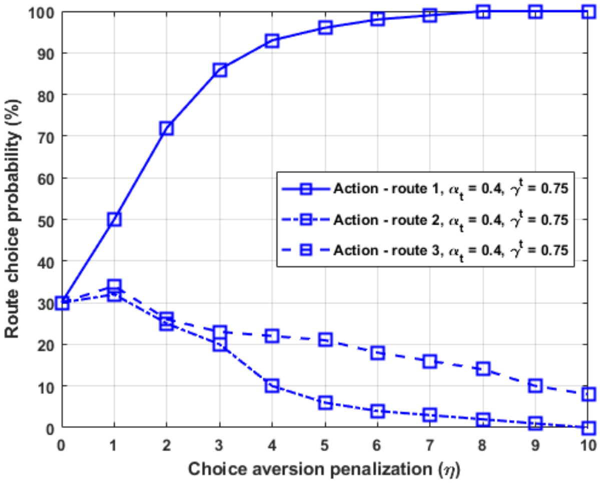


FIGURE 9. Route choice probabilities using the choice aversion model with the penalization factor, η.

E. EVALUATION OF ROUTE CHOICES USING CHOICE AVERSION

This section focuses on the analysis of node behavior in backhaul route selection in the presence of contextual information. In the case of route choices where uncertainty is due to variations in the actual travel times and information accuracy, the node's response is usually modeled by considering the utility maximization paradigm, which is affected by the risk perception. Depending on the considered traffic and policy, different behaviours such as risk aversion may be observed. The planning and acting of a node in a partially observable stochastic domain is evaluated. The route prediction probabilities generated by the RL-RDCM are evaluated as a function of an increasing value of the choice aversion penalization, η as shown in FIGURE 9 below:

As shown in FIGURE 9 above, as the value of η increases, the probability of packets being routed on route 1 increases. This means that the choice aversion model assigns more packets to route 1 because it has no alternative comparison in terms of minimum cost. This justifies the basic premise of prospect theory in environments with model uncertainty that agents tend to explore the route with minimum cost when they are reminded about the incremental cost of their actions. Applying cumulative prospect theory on the DRL strategy models the effect of the learning rate and noise level on the cumulative reward. The received stimulus is sampled from a normal distribution $\mathcal{N}(s, \sigma_s)$, thus the normal-distribution response is centred on the arithmetic means $\mu(s)$ of the normal probability distribution, and the width is determined by the arithmetic standard deviation σ_s. The cumulative rewards received based on IAB node response to selected actions under varying learning rates and noise levels are shown below:

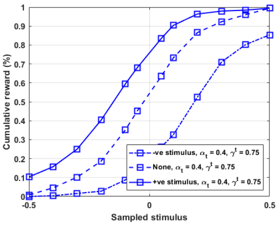


FIGURE 10. Noise level of 0.20 and learning rate $\alpha_t = 0.40$.

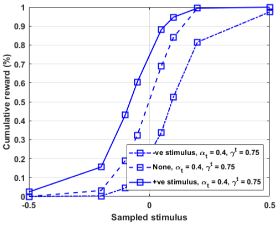


FIGURE 11. Noise level of 0.10 and learning rate $\alpha_t = 0.40$.

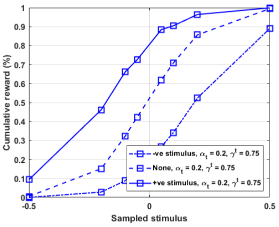


FIGURE 12. Noise level of 0.20 and learning rate $\alpha_t = 0.20$.

FIGURE 10 shows a sigmoid shape that is usually expected in cumulative probability functions. For the results in FIGURE 10, the learning rate, $\alpha_t = 0.40$ and the noise level is 0.20. The cumulative reward to the stimulus gradually increases initially, and it is larger for the positive stimulus. As the cumulative reward concaves up, it indicates an increase in new acquired information, and it nearly becomes linear indicating an approximately constant rate of acquiring knowledge. Then, the downward concave indicates the reduction in new information, meaning the system is no longer gaining new information, but using experience. The performance of route choice probability is evaluated with a reduced noise value of 0.10 and the result is shown in FIGURE 11. The results shown in FIGURE 11 indicate a rapid increase and a rapid decline in the rewards for the three routes. This means that the observed behavior of the cumulative reward is only due to the decrease in the noise level. Here, the upward concave of the graph is slow, indicating a delayed response to the stimulus. The performance of the scheme when the learning rate is reduced to $\alpha_t = 0.20$ and the noise value set as 0.20 is shown in FIGURE 12. In FIGURE 12, the effect of equaling the learning rate and noise level at 0.20 results in a more gradual increase and a more gradual decline in learning network information. The performance when the noise level is reduced to 0.10 and the learning rate remained at 0.20 is shown in FIGURE 13. A rapid increase and a rapid decline in the behavior of the cumulative reward is shown in FIGURE 13, similar to the response in FIGURE 11. The stimulus-reward results shown in the above figures all show deterministic responses and the use of the Metropolis-Hastings algorithm demonstrates the accuracy of route choices.

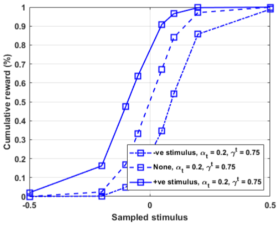


FIGURE 13. Noise level of 0.10 and learning rate $\alpha_t = 0.20$.

TABLE 3. The effect of the discount factor on the reward.

Discount factor, γ^t	Overall reward, $\hat{r}$	Most efficient path
0.75	99.99994034	0, 1, 3, 9, 10
0.80	99.99651029	0, 1, 3, 9, 10
0.85	99.98082485	0, 1, 3, 9, 10

F. EVALUATION OF CUMULATIVE THROUGHPUT

In this section, the cumulative reward in terms of the throughput of backhaul traffic with a varying learning rate, α_t. The performance of route establishment is evaluated using the Q-learning algorithm and the DNN architecture with the learning rate kept constant, i.e., $\alpha_t = 0.60$, while the value of the discount factor is increased. The effect of changing the values of the discount factor, γ^t, has been diagnosed and the results are shown in TABLE 3 below:

The results shown in TABLE 3 above were run over 1000 iterations, and the role of the discount factor is to determine how much the agent of the proposed algorithm cares about rewards in the distant future relative to those in the immediate future. A higher value of the reward was obtained for $\gamma^t = 0.75$ than when γ^t is increased. It was observed that when $\gamma^t \geq 0.85$, the sums do not converge for the policy, i.e., sums up to infinity. This means that at higher discount rates, the proposed algorithm becomes impulsive in choice behavior and does not show impulsive responses at lower discount factors. This raises a very important aspect that has always been ignored when AI strategies are applied in routing problems, which motivates a more interesting performance-complexity trade-off for IAB network design.

The effect of changing the value of γ^t is evaluated on route 1, i.e., the route with the highest choice probability. Additional analysis on the effect of the learning rate was conducted, and the relationship between the cumulative throughput and the number of iterations is shown in Fig. 14.

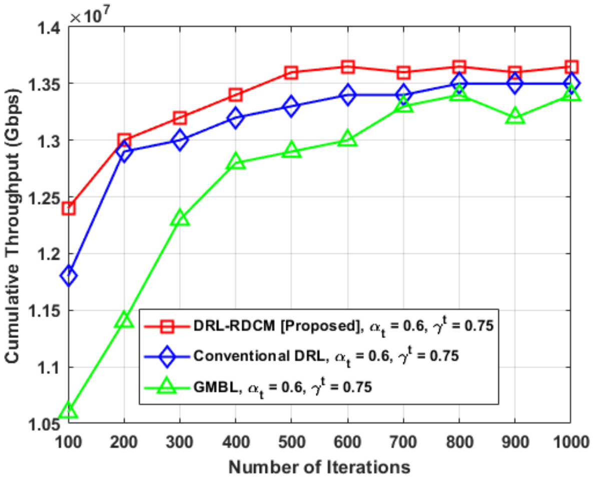


FIGURE 14. Cumulative throughput vs number of iterations with $\alpha_t = 0.6$ and $\gamma^t = 0.75$.

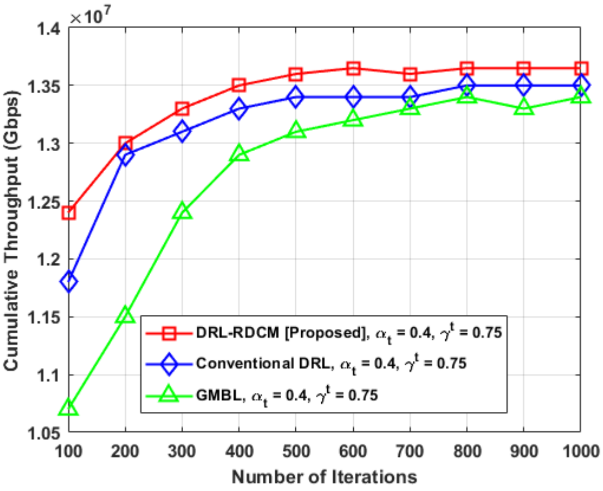


FIGURE 15. Cumulative throughput vs number of iterations with $\alpha_t = 0.4$ and $\gamma^t = 0.75$.

The cumulative throughput shown in Fig. 14 is for different learning algorithms, all using the same transmission budget. The transmission budget was set, and the cumulative throughput was evaluated for all the algorithms and the results show an increasing throughput trend for all the algorithms as the number of iterations increase. The proposed DRL-RDCM outperforms the conventional DRL and the GMBL by benefiting from the choice model used in its design. It must be noted that all the algorithms have a similar neural training, but differ in the RL agents they use. To test this relationship even further, the learning rate, α_t, is varied under the same behavior of the discount factor. The performance for $\alpha_t = 0.4$ is shown in Fig. 15 below:

The results shown in Fig. 15 above show that the oscillations caused by a large learning rate are reduced. The learning rate was further reduced to 0.2 and the performance is shown in Fig. 16. The simulation results suggest that reducing the learning rate helps the algorithm to learn better and it prevents the agent from being myopic and only learn actions that would produce immediate rewards. However, this is achieved at the cost of a large and increased time complexity. The action selection of the DRL strategy was shown to have similar complexity to the GMBL, the only difference being that the GMBL follows a procedure where each link is sampled a given number of times to determine its statistics to a desired level of accuracy. The resulting model is then used as an input to the CMDP framework and how much prediction error affects this adjustment also depends on the learning rate.

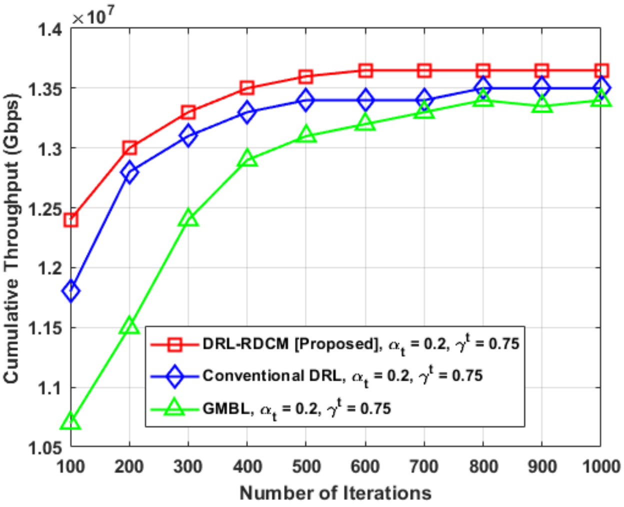


FIGURE 16. Cumulative throughput vs number of iterations with $\alpha_t = 0.2$ and $\gamma^t = 0.75$.

G. EVALUATING SYSTEM STABILITY BY INCORPORATING DELAY AND CONSTRAINTS

Up to this point, the system evaluation has exclusively focused on optimizing network utilities based on the transmission rates. Thus, the extension to this work is to incorporate the delay as a very important performance metric. Both the average throughput and the transmission delay were evaluated as the number of deployed IAB nodes between the traffic source and the donor node was increased. The end-to-end throughput was evaluated against the number of IAB nodes with a source rate of $R = 25$ Mbps and the result is shown in Fig. 17 below:

In Fig. 17 above, the throughput performance for the three algorithms is evaluated and an increasing trend is observed as the number of IAB nodes increase. The proposed DRL-RDCM strategy is observed to first lag the conventional DRL strategy, but as more spectrum becomes available in the network, it outperforms both baselines. This is because in as much as the RL strategy used in DRL is model-free, the use of the RDCM makes it behave more like a model-based strategy. This justifies the fact that quality improves as one moves away from the cell edge towards the center as capacity

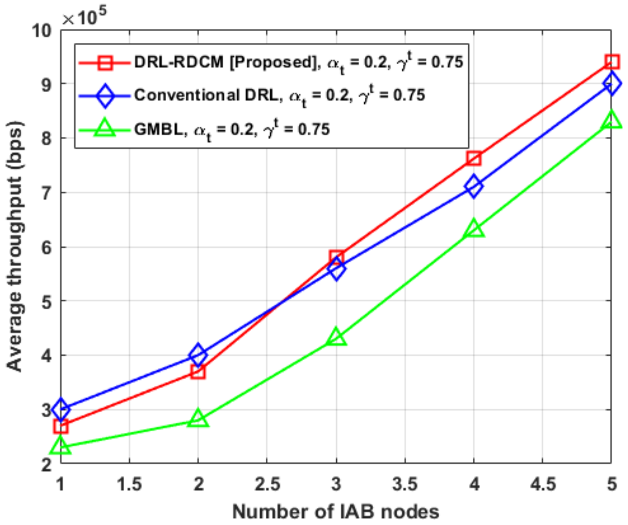


FIGURE 17. Average throughput vs. number of IAB nodes at $R = 25$ Mbps.

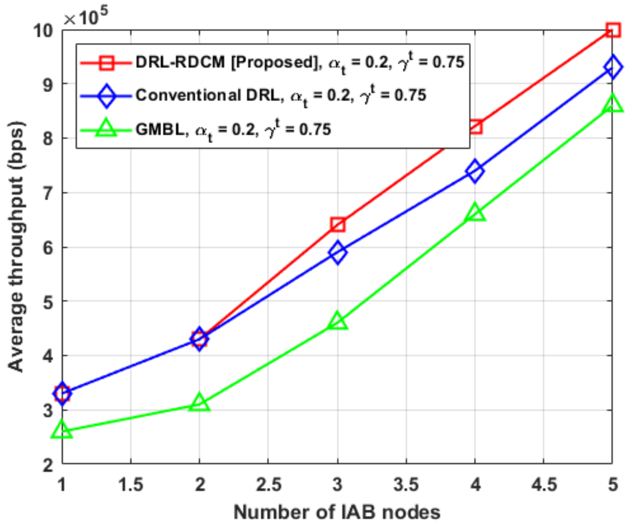


FIGURE 18. Average throughput vs number of IAB nodes at $R = 50$ Mbps.

becomes more guaranteed. The throughput performance is further evaluated with a source rate of $R = 50$ Mbps, and the results are shown in Fig. 18 below: Compared to the case of Fig. 17, in Fig. 18 above, a congested scenario was created by doubling the source rate and an overall improved performance of the three algorithms is observed, with the proposed DRL-RDCM performing much better than both baselines. The result shows that with a high source rate, the performance of the proposed algorithm progressively improves as the number of IAB nodes increase, which indicates that the proposed algorithm benefits more from coverage enhancement than the other two baselines. This indicates the strength of the proposed solution in terms of maximizing the backhaul link throughput without compromising the access QoS. The end-to-end latency for the configuration in Fig. 18 is shown in Fig. 19.

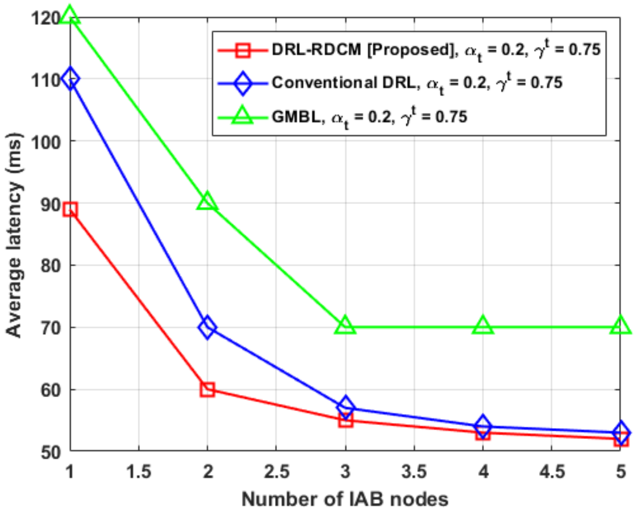


FIGURE 19. End-to-end latency vs number of nodes for $R = 50$ Mbps.

As expected, the average end-to-end latency decreases as the number of IAB nodes increases, as shown in Fig. 19. This is because as the number of IAB nodes increases, it provides more transmission routes, which decreases the average end-to-end delay of the system. From Fig. 19, it can be seen that the proposed DRL-RDCM provides significantly better end-to-end delay performance compared to the two baselines. This is particularly so for a small number of IAB nodes, and as the number of IAB nodes increases, the performance of the conventional DRL approach improves to closely follow that of the proposed scheme. On the other hand, the average latency for the GMBL approach remains constant at 70 ms as the number of IAB nodes increases to more than 3. This shows that the proposed algorithm adheres to reliable communication better than the other two baselines by better imposing the probabilistic delay constraint in (15). As expected, high throughput and lower transmission delays are achieved at the cost of high energy consumption, and the cost analysis of the proposed algorithm is considered in the following subsection.

H. EVALUATION OF THE COST FUNCTION

The evaluation of the power-delay trade-off as a function of the number of packets arriving at a node/link is the basic and underlying objective of wireless networks, and it cannot be overstated in IAB networks. Therefore, in this part this trade-off is evaluated in terms of: (i) the time delay of the learning process, (ii) the variation of mean delay and overflow costs with packet arrival rate, (iii) packet holding costs and power points as functions of packet arrivals.

1) TIME DELAY

The time complexity results in terms of populating useful attributes for the cost function are shown in TABLE 4.

From TABLE 4 above, it can be seen that populating the known attributes has higher time complexities than

TABLE 4. Time complexity.

Populated attribute	Attribute description	Elapsed time
Power cost		0.261440
Known transition probability function	From $s \to \tilde{s}$	0.573529
Buffer cost		0.447494
Unknown transition probability function	From $\tilde{s} \to s(t+1)$	0.013129
Unknown cost	From $\tilde{s} \to s(t+1)$	0.003936

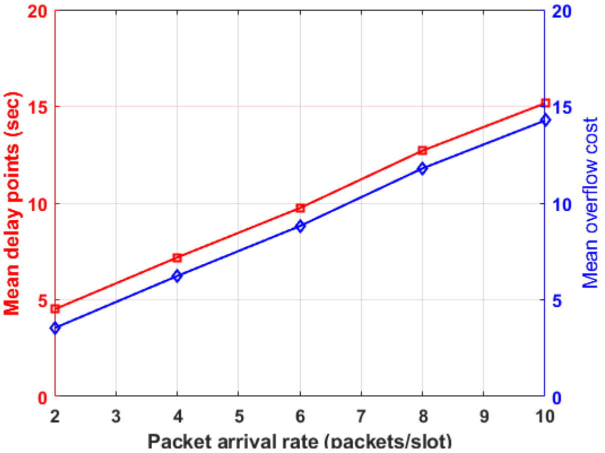


FIGURE 20. Mean delay points and mean buffer overflow costs vs packet arrival rates.

populating the unknown ones. This is because with the transition $s \to \tilde{s}$ the initial policy is not yet tuned to the specific traffic and channel conditions. The transition from $\tilde{s} \to s(t+1)$ the policy has already been tuned, hence less transition time is required.

2) DELAY POINTS - OVERFLOW COSTS

In this subsection, the cost function is evaluated using the post-decision state learning scheme. The overflow cost is actually the cost of delay, which is very crucial in agile network prioritization as it makes it possible for decision makers to consider the cost of keeping packets in the buffer beyond a single time slot. The performance of the system in terms of the mean delay and the mean overflow cost is evaluated as a function of an increasing packet arrival rate as shown in Fig. 20 below.

As shown in Fig. 20 above, both the mean delay and the mean buffer overflow cost increase linearly in a similar pattern as the packet arrival rate increases. In addition, the linear rate of increase of the two quantities is the same, which is about 1.25 per unit increase in packet arrival rate. The increase in the delay points pushes the Lagrange multiplier to its maximum, which results in the cost function weighting the delays more and the power consumption less. This leads to an increase in the buffer overflow cost as the system begins penalizing every packet held in the buffer more than necessary.

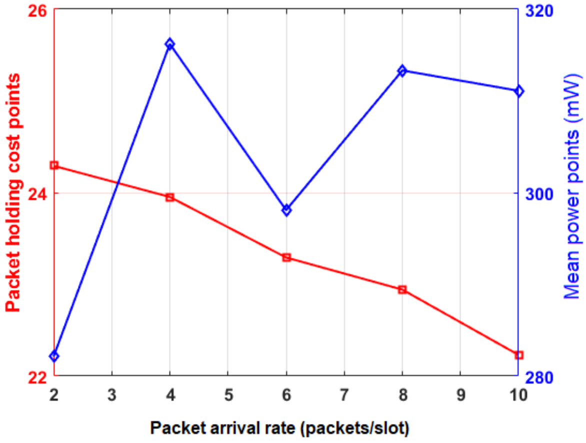


FIGURE 21. Packet holding cost points and mean power points vs packet arrival rate.

3) PACKET HOLDING COSTS - POWER POINTS

As the increasing delay forces the system to penalize any packet that is held in the buffer beyond a single time slot, the system will increase the transmission power in order to increase the transmission rate. This is expected to cause a decrease in the packet holding costs, however, at the cost of an increase in power consumption. This kind of behavior is verified in Fig. 21 below:

In Fig. 21 above, the system performance is evaluated in terms of the packet holding costs and power consumption as a function of increasing packet arrival rate. It can be observed that the packet holding cost points are decreasing, while the power consumption points increase. This is in line with the hypothesis that was made above, but there is a slight decrease in the packet holding cost points when the packet arrival rate increases from 4 to 6, which is followed by a sudden dip in power consumption. However, the power consumption suddenly increases when the packet arrival rate increases above 6 packets/slot. As the delay increases, the Lagrange multiplier is driven to its predefined maximum value, leading to an increase in power consumption, which then drives the packet holding costs down. This relationship indicates that in a system where packet losses are penalized, transmission delays result in non-Markovian system behaviour.

VII. CONCLUSION AND FUTURE WORK

This article proposed a backhaul adaptation scheme for IAB networks using DRL with RDCM in order to address the challenges of backhaul availability and backhaul scalability. The proposed scheme is controlled by the load on the access side of the network as well as the number of traffic flows being routed to the MBS. The problem is formulated as a CMDP and solved using a dual decomposition approach due to the existence of explicit and implicit constraints. A DRL strategy that takes advantage of an RDCM was then proposed and implemented. The advantage of the RDCM for this problem is that it incorporates choice aversion from prospect theory and the reward is not the only factor affecting the learning rates, but also the punishment. Optimal flow allocations in the network topology and the degree of aversion were derived using graph theory and RL. The Lagrange equivalent of the CMDP with respect to the policy and the punishment, and a cost function was also derived from the goodput distribution, and post-decision state learning was used to evaluate the power-delay cost trade-offs. The proposed algorithm was compared with the conventional DRL, i.e., without RDCM and GMBL algorithms, where it showed better throughput and delay performance over the two baselines. The near-optimal delay performance of the system is achieved before the optimal power consumption since the power consumption can only be learned after satisfying the packet holding cost constraint. The obtained results validate the objectives that were set out and outlined in Section I-C. It was observed that ML, in particular DRL, can be leveraged to improve throughput performance in mm-wave IAB networks, more especially by incorporating the RDCM.

A. PROS AND CONS OF THE PROPOSED ALGORITHM

In routing problems, it is beneficial to incorporate the concept of prospect theory that describes how decision makers choose between different prospects and how they estimate the perceived likelihood of each of these options. This is RL with foresight. In this work, the optimization objective was defined and the valuation function that was used was induced by an acceptance level through the RDCM for value functions that were specified in the prospect of route choices. These route choices have associated costs which are aimed at assisting in conflating observations in terms of alternatives leading to different choices, as well as rewards resulting from different choices of value function parameters from the characteristics of the CMDP. However, computational complexity is the main obstacle observed in the application of the proposed algorithm. It must be noted that the complexity of the family of RL strategies comes with the repeated learning updates towards reaching the reward. The proposed DRL-RDCM differs from the conventional DRL in terms of task complexity measures.

B. FUTURE RESEARCH WORK

Whether the system learns better by reward or by punishment, as well as to what extent does the reward and/or the punishment influence the learning rate of the system, have not been considered in RL-based IAB research solutions. However, it can be postulated, as an assumption, that the influence of the reward and punishment on the learning rate is subject to various complex mechanisms of the actions. For instance, the reward and punishment appear to be processed in different ways and the risk/loss aversion could also have an influence on the reward and punishment as well as algorithmic sensitivity to both the reward and punishment. It has been noted that few studies have investigated the influence of reward and punishment on learning rates, although this question has been addressed since the beginning of psychological research and is still unresolved in many aspects. Further research is not only required in the context of long-term effects (retention) of reward and punishment, but also whether reward or punishment lead to a higher learning rate and if so, under what conditions reward and punishment lead to higher learning rates.

For other future work, an approach that incorporates the RDCM with the GMBL algorithm could be developed and compared to the proposed scheme. In addition, the network model assumptions could be set to be the same as those used in the 3GPP study on IAB networks, for performance comparison with an established standard.

REFERENCES

[1] M. G. Kachhavay and A. P. Thakare, "5G technology-evolution and revolution," *Int. J. Comput. Sci. Mobile Comput.*, vol. 3, no. 3, pp. 1080–1087, Mar. 2014.

[2] 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.

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

[4] *NR; Study on Integrated Access and Backhaul*, document 38.874, 3GPP, Nov. 2018.

[5] C. Madapatha, B. Makki, C. Fang, O. Teyeb, E. Dahlman, M.-S. Alouini, and T. Svensson, "On integrated access and backhaul networks: Current status and potentials," *IEEE Open J. Commun. Soc.*, vol. 1, pp. 1374–1389, 2020.

[6] M. M. Ahamed and S. Faruque, "5G backhaul: Requirements, challenges, and emerging technologies," in *Broadband Communications Networks—Recent Advances and Lessons From Practice*. London, U.K.: IntechOpen, 2018. [Online]. Available: https://www.intechopen.com/chapters/62142, doi: 10.5772/intechopen.78615.

[7] O. Teyeb, A. Muhammad, G. Mildh, E. Dahlman, F. Barac, and B. Makki, "Integrated access backhauled networks," in *Proc. IEEE 90th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2019, pp. 1–5.

[8] L. Yanjun, L. Xiaobo, and Y. Osamu, "Traffic engineering framework with machine learning based meta-layer in software-defined networks," in *Proc. 4th IEEE Int. Conf. Netw. Infrastruct. Digit. Content*, Sep. 2014, pp. 121–125, doi: 10.1109/ICNIDC.2014.7000278.

[9] A. Valadarsky, M. Schapira, D. Shahaf, and A. Tamar, "Learning to route," in *Proc. 16th ACM Workshop Hot Topics Netw.*, Nov. 2017, pp. 185–191, doi: 10.1145/3152434.3152441.

[10] D. R. Militani, H. P. de Moraes, R. L. Rosa, L. Wuttisittikulkij, M. A. Ramirez, and D. Z. Rodriguez, "Enhanced routing algorithm based on reinforcement machine learning—A case of VoIP service," *Sensors*, vol. 21, no. 2, pp. 1–32, Jan. 2021.

[11] T. K. Vu, C.-F. Liu, M. Bennis, M. Debbah, and M. Latva-Aho, "Path selection and rate allocation in self-backhauled mmWave networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2018, pp. 1–6.

[12] 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.

[13] Q. Cheng, Z. Wei, and J. Yuan, "Deep reinforcement learning-based spectrum allocation and power management for IAB networks," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2021, pp. 1–6.

[14] R. Cole, Y. Dodis, and T. Roughgarden, "Bottleneck links, variable demand, and the tragedy of the commons," *Networks*, vol. 60, no. 3, pp. 194–203, Oct. 2012.

[15] M. Sniedovich, "Dijkstra's algorithm revisited: The dynamic programming connexion," *J. Control Cybern.*, vol. 35, no. 3, pp. 599–620, 2006.

[16] S. M. Azimi-Abarghouyi, B. Makki, M. Haenggi, M. Nasiri-Kenari, and T. Svensson, "Coverage analysis of finite cellular networks: A stochastic geometry approach," in *Proc. Iran Workshop Commun. Inf. Theory (IWCIT)*, Apr. 2018, pp. 1–5.

[17] B. Zhang, F. Devoti, I. Filippini, and D. De Donno, "Resource allocation in mmWave 5G IAB networks: A reinforcement learning approach based on column generation," *Computer Netw.*, vol. 196, Jun. 2021, Art. no. 108248.

[18] J. Hyun, N. Van Tu, and J. W.-K. Hong, "Towards knowledge-defined networking using in-band network telemetry," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Apr. 2018, pp. 1–7.

[19] M. B. Dagues, C. L. Hall, and L.-A. Giraldeau, "Individual differences in learning ability are negatively linked to behavioural plasticity in a frequency-dependent game," *Animal Behav.*, vol. 159, pp. 97–103, Jan. 2020.

[20] S. Jouet, C. Perkins, and D. Pezaros, "OTCP: SDN-managed congestion control for data center networks," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Apr. 2016, pp. 171–179.

[21] M. D. Hill, "Three other models of computer system performance," 2019, *arXiv:1901.02926*.

[22] S. Samarakoon, M. Bennis, W. Saad, and M. Debbah, "Distributed federated learning for ultra-reliable low-latency vehicular communications," *IEEE Trans. Commun.*, vol. 68, no. 2, pp. 1146–1159, Nov. 2019.

[23] C. Xu, "A deep reinforcement learning approach for software-defined networking routing optimization," in *Proc. 4th Int. Conf. Comput. Sci. Appl. Eng.*, 2020, pp. 1–5.

[24] Y. Liu, J. Ding, and X. Liu, "IPO: Interior-point policy optimization under constraints," in *Proc. AAAI Conf. AI*, Apr. 2020, vol. 34, no. 4, pp. 4940–4947.

[25] E. Pan, P. Petsagkourakis, M. Mowbray, D. Zhang, and E. A. D. Rio-Chanona, "Constrained model-free reinforcement learning for process optimization," *Comput. Chem. Eng.*, vol. 154, pp. 107462–107480, Nov. 2021.

[26] T. Ai, V. Wijeratne, and A. A. Wahid, "Impact of buffer sizing on energy efficiency and performance," *IET Netw.*, vol. 4, no. 1, pp. 1–9, 2015.

[27] 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.

[28] Q. Xu and J. Sun, "A simple active queue management based on the prediction of the packet arrival rate," *J. Netw. Comput. Appl.*, vol. 42, pp. 12–20, Jun. 2014.

[29] M. Zimmermann and E. Frejinger, "A tutorial on recursive models for analyzing and predicting path choice behavior," *EURO J. Transp. Logistics*, vol. 9, no. 2, pp. 1–12, Jun. 2020.

[30] F. L. Lec and B. Tarroux, "On attitudes to choice: Some experimental evidence on choice aversion," *J. Eur. Econ. Assoc.*, vol. 18, no. 5, pp. 2108–2138, Oct. 2020.

[31] D. Fudenberg and T. Strzalecki, "Dynamic logit with choice aversion," *Econometrica*, vol. 83, no. 2, pp. 651–691, 2015.

[32] A. Hansen, "The three extreme value distributions: An introductory review," *Frontiers Phys.*, vol. 8, pp. 1–8, Dec. 2020.

[33] Q. Liang and E. Modiano, "Optimal network control in partially-controllable networks," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2019, pp. 397–405.

[34] E. Cristiani, C. De Fabritiis, and B. Piccoli, "A fluid dynamic approach for traffic forecast from mobile sensors data," *Commun. Appl. Ind. Math.*, vol. 1, pp. 54–71, Jun. 2010.

[35] A. Morales and M. Villapol, "Reviewing the service specification of the IEEE 802.16 MAC layer connection management: A formal approach," *CLEI Electron. J.*, vol. 16, no. 2, pp. 1–12, Aug. 2013.

[36] T. K. Vu, C. F. Liu, M. Bennis, M. Debbah, M. Latva-Aho, and C. S. Hong, "Ultra-reliable and low latency communication in mmWave-enabled massive MIMO networks," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2041–2044, Sep. 2017.

[37] L. G. Afanaseva and S. A. Grishunina, "Stability conditions for a multiserver queueing system with a regenerative input flow and simultaneous service of a customer by a random number of servers," *Queueing Syst.*, vol. 94, nos. 3–4, pp. 213–241, Apr. 2020.

[38] 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.

[39] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 1, Oct. 2014, pp. 1–9.

[40] K. Fredenslund. (Mar. 2018). *Computational Complexity of Neural Networks*. Accessed: Mar. 22, 2022. [Online]. Available: https://lunalux.io/series/introduction-to-neural-networks/computational-complexity-of-neural-networks

[41] 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, 2019.

[42] T. Lattimore, "The sample-complexity of general reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 28–36.

[43] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Dec. 2019, pp. 11918–11930.

[44] A. Agarwal, S. Kakade, and L. F. Yang, "Model-based reinforcement learning with a generative model is minimax optimal," in *Proc. 33rd Conf. Learn. Theory*, vol. 125, 2020, pp. 67–83.

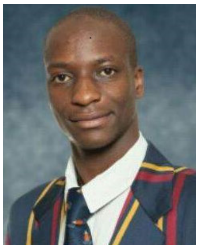
[45] A. C. M. Echeverria, S. Schopohl, A. Taalaibekova, and V. Vannetelbosch, "Coordination on networks with farsighted and myopic agents," *SSRN Electron. J.*, vol. 51, pp. 509–536, Jan. 2022.

[46] M. M. Sande, M. C. Hlophe, and B. T. Maharaj, "Access and radio resource management for IAB networks using deep reinforcement learning," *IEEE Access*, vol. 9, pp. 114218–114234, 2021.

[47] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using NetworkX," in *Proc. 7th Python Sci. Conf.*, 2008, pp. 11–16.

[48] J. F. D. Valgas, J. F. Monserrat, and H. Arslan, "Flexible numerology in 5G NR: Interference quantification and proper selection depending on the scenario," *Mobile Inf. Syst.*, vol. 2021, pp. 1–9, Mar. 2021.

[49] W. Su, L. Chen, M. Wu, M. Zhou, Z. Liu, and W. Cao, "Nesterov accelerated gradient descent-based convolution neural network with dropout for facial expression recognition," in *Proc. 11th Asian Control Conf. (ASCC)*, Dec. 2017, pp. 1063–1068.



MALCOLM M. SANDE (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 with the Sentech Chair in Broadband Wireless Multimedia Communications (BWMC) Research Group. His research interest includes the application of machine learning techniques in mobile and wireless communications, with 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 the Broadband Wireless Multimedia Communications (BWMC) Group, 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, predictive analytics and inference with applications in wireless communications, finance, health, and robotics.

BODHASWAR T. SUNIL MAHARAJ (Senior Member, IEEE) received the Ph.D. degree in engineering in the area of wireless communications from the University of Pretoria. He is currently a Full Professor and holds the research position at the Sentech Chair in 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.