sensors

MDPI

Article

Multi-Objective Optimization of Energy Saving and Throughput in Heterogeneous Networks Using Deep Reinforcement Learning

Kyungho Ryu and Wooseong Kim *

Department of Computer Engineering, Gachon University, Seongnam 13120, Korea; rudgh1368@gachon.ac.kr
* Correspondence: wooseong@gachon.ac.kr

Abstract: Wireless networking using GHz or THz spectra has encouraged mobile service providers to deploy small cells to improve link quality and cell capacity using mmWave backhaul links. As green networking for less CO_2 emission is mandatory to confront global climate change, we need energy efficient network management for such denser small-cell heterogeneous networks (HetNets) that already suffer from observable power consumption. We establish a dual-objective optimization model that minimizes energy consumption by switching off unused small cells while maximizing user throughput, which is a mixed integer linear problem (MILP). Recently, the deep reinforcement learning (DRL) algorithm has been applied to many NP-hard problems of the wireless networking field, such as radio resource allocation, association and power saving, which can induce a near-optimal solution with fast inference time as an online solution. In this paper, we investigate the feasibility of the DRL algorithm for a dual-objective problem, energy efficient routing and throughput maximization, which has not been explored before. We propose a proximal policy (PPO)-based multi-objective algorithm using the actor-critic model that is realized as an optimistic linear support framework in which the PPO algorithm searches for feasible solutions iteratively. Experimental results show that our algorithm can achieve throughput and energy savings comparable to the CPLEX.

Keywords: wireless heterogeneous network; energy saving; wireless backhaul mesh; deep reinforcement learning

Citation: Ryu, K.; Kim, W. Multi-Objective Optimization of Energy Saving and Throughput in Heterogeneous Networks Using Deep Reinforcement Learning. *Sensors* **2021**, *21*, 7925. https://doi.org/10.3390/s21237925

Academic Editors: Gianmarco Romano, Peter Han Joo Chong and Omprakash Kaiwartya

Received: 10 October 2021
Accepted: 19 November 2021
Published: 27 November 2021

1. Introduction

Exponentially increasing mobile traffic accelerates the deployment of dense small cells operating on the 3 GHz spectrum under legacy macro cells, called a heterogeneous small cell network (HetNet), which offloads congested macro cells and eventually enhances quality of user experience (QoE). User equipments (UEs) can have dual connectivity to the macro eNB (MeNB) and small eNB (SeNB) for control/data bearer splitting or download busting. Such SeNB deployment is costly when backhauling to a network gateway (a MeNB in this paper). Millimeter-wave (mmWave)-based backhauling can reduce deployment efforts and provide gigabit data rates to UEs using huge bandwidths, such as 9 and 10 GHz, available at the 60 GHz band and E-band. Many measurement campaigns and demonstrations at 28, 38, 60 and 73 GHz have already shown the feasibility of mmWave use for mobile communication [1–3].

To overcome the short communication range of the mmWave link due to its high pathloss and low penetration, beam forming based on directional antennae and repeaters for amplifying is necessarily considered. Figure 1 shows the HetNet equipped by a multi-hop backhaul mesh network for long-range backhauling of the mmWave links, in which an SeNB unreachable by the MeNB can access the Internet through multi-hop relays of the SeNBs [4,5]. The mmWave-based backhaul mesh networks have several challenges, such as efficient radio resource management (RRM) [5,6], interference management [7,8], multi-hop routing [9], and energy saving [10].

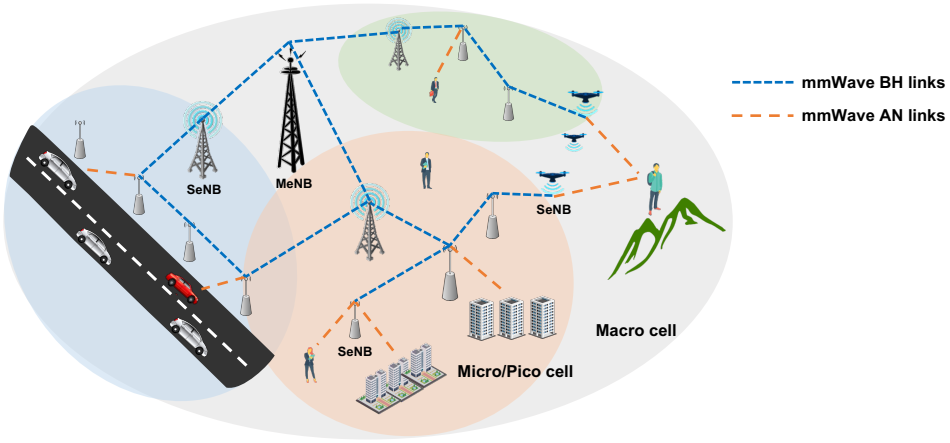


Figure 1. Heterogeneous cellular network architecture with mmWave backhaul mesh.

Due to increasing power consumption from excessively deployed SeNBs and mmWave backhaul transmissions, various approaches to save energy in mobile networks have been considered [11]; these include switching off small and macro cells [12–14] or adjusting cell size dynamically [15,16], where users of switched-off SeNBs are supported by neighboring SeNBs using remaining resources.

Especially for the HetNets with mmWave-based backhauls, Chen et al. [17] introduced a user association and power allocation algorithm for energy harvesting and self-backhaul SeNB to maximize energy efficiency. Additionally, Mesodiakaki et al. [18] studied an energy- and spectrum-efficient user association problem considering mmWave backhauls. Hao et al. [19] investigated the energy-efficient resource allocation in two-tier massive multiple-input multiple-output (mMIMO) HetNets with wireless backhauls.

Most previous works focus on radio resource allocation to increase spectral and energy efficiency in the HetNets. However, in the mmWave backhaul mesh, a multi-hop routing mechanism determines energy saving, as SeNBs need to be switched on for relaying regardless of the presence of associated users. We establish a fluid model of user traffic in the mmWave-based backhaul mesh and solve the joint optimization problem that minimizes energy consumption while guaranteeing the demanded data rate of each UE [9]. This problem can be formulated in a non-convex mixed integer linear problem (MILP), known as a NP-hard. When we used the branch-and-cut algorithm of CPLEX to find an optimum in a given HetNet topology, it consumed more than 30 min of calculation time, which is infeasible, as the HetNet topology changed dynamically due to UE mobility. For the online algorithm, previous works [17–19] considered heuristic or iterative algorithms, which cannot be guaranteed to find a near optimal solution or can suffer from convergence delays. In this study, we consider a deep reinforcement learning (DRL) algorithm to find a feasible solution of the MILP problem in real time.

Reinforcement learning (RL) [20] has received much attention for dynamic systems, which can provide a long-term solution considering future rewards. Furthermore, the deep learning technique has recently been applied to overcome the curse of dimensionality as the size of the Markov decision process (MDP) increases in terms of state and action space [21–25]. RL based on a deep neural network (DNN) can provide a feasible online solution; feed-forward computation is simple for inference compared to backward computation for training. Thus, many researchers now consider the DRL algorithm to solve NP-hard problems of the wireless communication and networking field.

Recently, many studies about applying DRL to wireless communication problems have been introduced, as in the related work section. Several works [26–37] used DRL to allocate radio resources, transmission power and channels to increase spectral efficiency; additionally, multiple access schemes were also exploited by DRL in [38–41]. For energy saving, several studies developed a DRL algorithm for an energy-efficient multi-hop routing protocol or peer-to-peer connectivity in the ad hoc networks of satellites or UAVs [42,43],

where individual mobile agents learn an optimal policy to maintain connectivity while saving limited power. [44–46] introduced energy-saving mechanisms using DRL, wherein an agent controls the transmission power, association and sleep mode of SeNBs in a HetNet without multi-hop backhauls. To the best of our knowledge, this is the first work that investigates DRL to find the Pareto front of a multi-objective optimization problem of energy saving and throughput maximization in the HetNet with an mmWave-based multi-hop backhaul mesh.

Key motivations of this study are enumerated as below:

- There has not been notable research on an energy efficient multi-hop routing algorithm using DRL for an mmWave backhaul mesh of a dense HetNet;
- The DRL-based algorithm can be considered to find a Pareto front solution for the dual-objective optimization of energy saving and throughput maximization in the HetNet.

To solve our optimization problem, we adopt a proximal policy optimization (PPO)-based DRL algorithm [24] which shows typically fast and reliable convergence in the training phase as one of popular policy-based DRL algorithms. The PPO algorithm can provide an online policy for controlling backhaul transmission and SeNB power in HetNets, and it is simple to implement but comparable with the complicated trust region policy optimization (TRPO) [23] in terms of performance. However, it is a challenge for the PPO algorithm to find an optimum of the multi-objective problem if only the reward sum of conflicting multi-objectives is given to an agent for training. Therefore, we consider a multi-objective reinforcement learning (MORL) approach [47] to find the Pareto front solutions.

Optimistic linear support (OLS) is proposed for the MORL [48], in which an outer loop iteratively calls a single-objective solver based on the deep Q-network as a subroutine. In this paper, we propose PPO-based deep optimistic linear support (PDOLS), where the PPO algorithm iteratively solves the scalarized objective problem by a specific weight vector for rewards. In experiments, the proposed PDOLS searched optimal corner weights for multi-objectives efficiently and resulted in similar outcomes to the optimal weights obtained through repeated experiments. Additionally, the PDOLS achieved notable throughput and energy saving compared to the CPLEX results [9]; the CPLEX achieves a 35% energy savings and a 14 Mbps data rate without blockage, while the PDOLS achieves an almost 28% energy savings and a 13.4 Mbps data rate. Such performance reduction is small, considering the CPLEX execution time and DRL inference time are 30 min vs. 1 s. Furthermore, we improve the PDOLS with a scaled reward (PDOLS-SR) that adjusts the reward values according to the environment, which increases the probability of finding the optimal weight vector.

We highlight our key contributions of this study as below:

- We propose a PPO-based online algorithm for the bi-objective problem of energy minimization and throughput maximization;
- We propose an integrated framework based on the PPO algorithm and OLS to find the Pareto front of the two objectives;
- We demonstrate the feasibility of the proposed online solution based on DRL in a HetNet environment.

The remainder of the paper is organized as follows. We introduce recent works on DRL for wireless networking solutions in Section 2, and offer an overview of the DRL background in Section 3. In Section 4, we establish the multi-objective optimization model for energy saving and throughput maximization in HetNets. We propose the PPO and PDOLS algorithm for the multi-objective optimization problem in Section 5. Section 6 shows our experimental results regarding performance of the learning algorithm and HetNet throughput. Finally, we discuss and conclude our study in Section 7.

2. Related Works

Previously, most of the NP problems in the wireless communication and networking area were solved by linear approximation or heuristic algorithms, such as simulated annealing (SA), generic algorithm (GA), particle swarm optimization (PSO), etc. Recent successes

of the DNN technique in computer vision and speech recognition show the possibility of applying large-scale feed-forward neural networks to wireless networking. Therefore, the 1D or 2D convolution neural network (CNN) that is popular for computer vision and image processing was used for wireless channel estimation with MIMO [49–51], automatic modulation and coding schemes [52–54] and network intrusion detection [55–58].

In contrast to the above supervised deep learning, artificial intelligence for controlling dynamics of the wireless networking system needs to be made naturally by past experience in the system. Such dynamic systems can be modelled by the MDP; at each step, a network agent acts based on the state and receives reward feedback for the action, such as successful transmission, packet loss, collision, saving power, etc. Using the collected experience data, the DRL algorithm can effectively find an optimal solution of the wireless networking system. The following studies have demonstrated feasibility of using DRL algorithms for wireless communication and networking during the last several years (refer to the summary in Table 1).

Table 1. DRL-empowered wireless communication and networking research.

References	Areas of DRL Studies on Wireless Communications
[26–32]	Cognitive radio and dynamic wireless channel selection increase spectral efficiency, which is typically a combinatoric problem of matching channels to nodes. Using DRL, agents can learn the optimal policy from the degree of interference as a reward for every action of channel selection.
[38–41]	The wireless link layer provides a media access scheme for multiple users which is realized in a MAC protocol. Several studies design the wireless MAC protocol based on the DRL algorithm, in which DRL agents learn an optimal transmission policy from the reward of contention resolution at a particular channel state.
[59–61]	A user association or handover algorithm for a serving base station affects throughput and QoS of each user. The DRL algorithm enables UEs to select an optimal base station based on past experience.
[33–37]	Wireless networks have various resources to be scheduled, such as radio block, channels, sequence codes, power, time slots, etc. Many of the scheduling problems have non-convex feasible set and user mobility, which makes the problems intractable. The DRL agents learn an optimal scheduling policy repeatedly from resource utilization against a chosen allocation.
[42–44,62,63]	Energy and power consumption is critical, especially for green wireless networking, mobile edge cloud networks and UAV networks. The DRL algorithm explores possible policies based on the reward of energy saving while guaranteeing throughput constraint.

Wang et al. [26] proposed a dynamic multi-channel access mechanism based on deep Q-learning. A node selects one multi-channel that has low interference, which returns the maximum reward for the action. Zhong et al. [27,28] used the actor-critic algorithm to explore the sensing policy for dynamic channel access and considered a multi-agent model for distributed sensors in a partially observable environment. Naparstek et al. [29,30] also proposed DQN-based multi-agents which act based on Q-value independently. Li et al. [31] applied the DQN for channel sensing, and Liu et al. [32] proposed a hierarchical deep Q-network (h-DQN) model for cooperative channel sensing, which divides the original problem into separate sub-problems for multi-DRL agents.

Ali et al. [38] introduced a Q-learning-based MAC protocol in dense WLANs which learns the optimal policy based on channel state and transmission action experience. Yu et al. [39] investigated a DRL-based MAC protocol for heterogeneous wireless networking which was called deep-reinforcement learning multiple access (DLMA). They established a new multi-dimensional RL framework based on the Q-learning that maximizes sum throughput and provides proportional fairness, even co-existing with TDMA-like ALOHA protocols. Al et al. [40] studied radio resource scheduling (RRS) in the cellular MAC layer using the DQN. Nisioti et al. [41] presented a MAC solution for sensor networks based on coordinated reinforcement learning by considering the dependencies among sensors to find the optimal actions.

Zhao et al. [59] studied user association and radio resource allocation in a HetNet. For a large action space, they considered a multi-agent RL approach and a dueling double deep Q-network (D3QN) to obtain an optimal policy with little computation complexity. Zhang et al. [60] proposed a DRL algorithm for the association between each IoT device and a cellular user to maximize the sum rate of all the IoT devices in symbiotic radio networks (SRNs). Ding et al. [61] introduced the user association and power control scheme using the multi-agent DQN to ensure the UE's quality of service (QoS) requirements.

He et al. [33] proposed an orchestration framework in vehicular networks with a novel DRL algorithm for the resource allocation of networking, caching and computing resources. Shi et al. [34] modelled a hierarchical DRL-based multi-DC (drone cell) trajectory planning and resource allocation scheme for high-mobility users. In [35,36], the authors also conducted resource allocation for uplink nonorthogonal multiple access (NOMA) systems using a DRL-based algorithm to solve the nonconvex optimization problem. Rahimi et al. [37] also tried to increase scalability with a hierarchical DRL for joint user association and resource allocation in the NOMA system.

Liu et al. [43] introduced a novel DRL-based energy-efficient routing protocol called DRL-ER, which avoids the battery energy imbalance of constellations and guarantees a required end-to-end delay bound. Liu et al. [42] adopted a DRL-based energy-efficient control for coverage and connectivity in UAV communication systems. Du et al. [62] reviewed and analyzed how to achieve green DRL for radio resource management (RRM). Dai et al. [63] utilized DRL to design an optimal computation offloading and resource allocation strategy for minimizing energy consumption. El et al. [44] solved the energy-delay-trade-off (EDT) problem in a HetNet where small cells can switch to different sleep mode levels to save energy while maintaining QoS using the DRL.

To the best of our knowledge, our study is first to develop a PPO-based multi-objective algorithm that controls multi-hop routing and switching on/off SeNBs in HetNets, even though many previous works have applied the DRL algorithm for other optimization problems.

3. Deep Reinforcement Learning (DRL)

This section provides a brief overview of reinforcement learning (RL) and DRL. RL is a popular machine learning algorithm which allows agents to learn optimal behavior through trial-and-error interactions with a dynamic environment. A key strategy of the RL is utilizing statistics to obtain an optimal control decision (policy) in the form of the MDP. The MDP is modelled by (S, A, P^a_{ss}, R^a), wherein the state space is represented by S, the action space is represented by A, the state transition probability is $P_{ss'}$ at a taken action a and a corresponding reward R, and in which the policy as a function $\pi(s)$ specifies an action a in each state s. Therefore, an optimal policy, π^*, maximizes the expected reward for future T steps, $\mathbf{E}[\sum_{t=0}^{T} \gamma^t r_t]$, where γ is a discount factor ($0 \leq \gamma < 1$) for the infinite-horizon discounted model.

For effective agent learning, the estimation of a state-value function for a state s is critical; $V_\pi(s) = \mathbf{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^t R(s_{t+k+1}) \middle| S_t = s \right]$ at a time step t. Additionally, suppose that a certain action, a, is taken in the state s; then, an action-value Q-function can be defined as $q_\pi(s,a) = \mathbf{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^t R^a(s_{t+k+1}) \middle| S_t = s, A_t = a \right]$. According to the Bellman optimality equation, the optimal value function, $V_*(s)$, can be decomposed recursively as $V_*(s) = \max_a \mathbf{E}[R_{t+1} + \gamma V_*(s_{t+1}) | S_t = s, A_t = a]$, which tells us that the expected return from the best action is the same as the state value of an optimal policy.

3.1. Deep Q-Learning

As the state and action spaces become larger and continuous, function approximation is mandatory for Q-learning instead of using a legacy tabular form of actions and Q-values. Although the combination of RL and neural networks was considered a long time ago,

it is only very recently that DRL algorithms based on deep neural networks (DNNs) has received much attention instead of the linear function approximation [20,64]. DNNs represent a function with higher complexity by employing a deep hierarchical layer architecture that constitutes a non-linear information processing unit. Deep learning approximates such a mapping function for statistical curve fitting with labeled training datasets.

The DRL utilizes the training process of the DNN based on datasets which can improve learning speed and performance without the MDP model information (the R and $P_{ss'}$ are unknown). The DRL induces a policy based on a value function, $V_\pi(s)$, approximated by the DNN, which is trained using the batch of samples (S, A, R, S') that an agent collects by interacting with the environment. In a sequence of discrete time, $\{t = 0, 1, 2, \ldots\}$, the agent selects an ε-greedy action for the maximum reward given by $V_\pi(s)$; the ε provides randomness to explore and avoid the local minimum.

Mnih et al. introduced the deep Q-network (DQN) in [22], which is a seminal work for Q-function approximation based on DNNs. In particular, they addressed and solved two challenges in the DRL; first, the deep learning assumes that the data samples are iid (independent identically distributed), but actually the next state, s', is correlated with the current state, s, in the MDP. Second, the target model for training is non-stationary, as the model parameters θ are updated at every iteration. For this, the DQN adopts an experience-replay buffer for the training and separation of the main and target networks. The DQN updates θ of the main network by minimizing temporal-difference errors, $L(\theta) = Y_t - Q(s_t, a_t; \theta)$, where $Y_t = r_t + \gamma \cdot max_{a'} \hat{Q}(s_{t+1}, a'; \theta^-)$ and the state-action value function, $Q(s, a; \theta)$ are given by the target and main network, respectively. The target network is periodically updated by the main network.

3.2. Policy Gradient and Actor-Critic

The DQN is limited to high dimensional and continuous action spaces that demand iterative optimization processes at every step. Additionally, discretizing the continuous action values cannot avoid the curse of dimensionality due to a large number of actions, or, probably, loses important information of the action space from quantization.

Therefore, the policy gradient (PG) algorithm is used mostly for high dimensional and continuous actions [65,66], which adjusts the model parameter, θ, of a policy function in the direction of the stochastic policy gradient (SPG), $\nabla_\theta J(\pi_\theta)$.

$$\begin{aligned} \nabla_\theta J(\pi_\theta) &= \int_S \rho^\pi(s) \int_A \nabla_\theta \pi_\theta(a|s) Q^\pi(s, a) \, da ds \\ &= \mathbf{E}_{s \sim \rho^\pi, a \sim \pi_\theta} [\nabla_\theta \pi_\theta(a|s) Q^\pi(s, a)] \end{aligned} \tag{1}$$

The PG algorithm [21] can be implemented by the actor-critic architecture, in which the actor stochastically updates the θ of the policy function while the critic evaluates the policy and updates the action-value function approximator, $Q^w(s, a)$, in such a direction as to minimize error, $\epsilon^2(w) = \mathbf{E}_{s \sim \rho^\pi, a \sim \pi_\theta}[(Q^w(s, a) - Q^\pi(s, a))^2]$. As the dimension of action spaces increases, deterministic policy gradient (DPG) as a special case of the SPG is efficient to derive only the mean of the state spaces compared to the SPG, $\lim_{\sigma \downarrow 0} \nabla_\theta J(\pi_{\mu_\theta, \sigma}) = \nabla_\theta J(\mu_\theta)$.

4. System Model

In this section, we establish a mathematical system model of the HetNet with a mmWave backhaul mesh among SeNBs and MeNBs in which energy consumption and user traffic for the mmWave backhaul links and access links are formulated. In this model, we present dual objectives to minimize the energy while maximizing the user throughput. The symbols used in this model are described in Table 2.

Table 2. Parameters (P) and variables (V) used in the model.

Symbol	Description	
B_{RB}	Bandwidth for a RB	P
c_{ij}	Maximum capacity of link (*i*,*j*)	P
C_i^{max}	Maximum AN capacity of eNB *i*	P
e_i	Total energy consumption at node *i*	V
f_{ij}^u	Flow of UE *u* on link (*i*,*j*)	V
x_{ij}^u	Indicator if UE *u* uses link (*i*,*j*)	V
$\mathcal{I}$	Set of interference links	P
$\mathcal{L}$	Set of links	P
$\mathcal{L}_{\mathcal{AN}}$	Set of AN links	P
$\mathcal{L}_{\mathcal{BH}}$	Set of BH links	P
$\mathcal{N}$	Set of eNB	P
$\mathcal{M}$	Set of Macro eNB (MeNB)	P
$\mathcal{S}$	Set of Small eNB (SeNB)	P
$\mathcal{U}$	Set of UE	P
$N_{a_{iu}}$	Number of antennas (MIMO) for UE *u* at eNB *i*	P
N_{RB_i}	Number of RBs at node *i*	P
P_{0_i}	Static power at node *i*	P
R^u	User demand data rate *u*	V

4.1. Energy Consumption Model

The energy consumption of eNB *i* is composed of two folds: energy consumption from access links toward UEs and backhaul links toward other eNBs,

$$e_i = e_i^{AN} + e_i^{BN}, \tag{2}$$

where energy consumption in the access network (AN) and backhaul network (BN) are e_i^{AN} and e_i^{BN}, respectively.

4.1.1. AN Energy Consumption

According to the linear approximation [67] between relative RF output power and the power consumption of an eNB, energy consumption for the access links can be derived as

$$e_i^{AN} = P_{0_i}^{AN} + \Delta_p \cdot P_{out_i}^{AN} \qquad \forall i \in \mathcal{N} \tag{3}$$

where Δ_p is a multiplier for load-dependent power consumption, which is different from the type of antenna (refer to Table 3) [67].

$$P_{out_i}^{AN} = P_{max_i}^{AN} \cdot F_i^{AN} = P_{max_i}^{AN} \cdot \frac{1}{N_{RB_i}} \sum_{u \in \mathcal{U}} \left\lceil \frac{f_{iu}^u x_{iu}^u}{N_{a_{iu}}^{AN} \cdot B_{RB} \cdot log_2(1 + SINR_{iu})} \right\rceil \tag{4}$$

where the SINR is the signal-to-noise and interference ratio, $P_{out_i}^{AN}$ is the power consumption of the transceiver for the access links for all associated UEs, and $0 < P_{out_i}^{AN} \leq P_{max_i}^{AN}$. $P_{max_i}^{AN}$ is the maximum transmission power for the AN transceiver at the eNB *i*. The $P_{out_i}^{AN}$ can be scaled by the aggregated flow rate F_i^{AN} against the link capacity, which is the same as the ratio of radio resource blocks (RB) used by all associated UEs to the total available RBs (N_{RB_i}); the number of used RBs can be calculated by dividing the sum of user data rate by the rate of a single RB (bandwidth B_{RB} Hz). $N_{a_{iu}}^{AN}$ is the number of antenna for MIMO and f_{iu}^u is the data rate for each UE. x_{iu}^u is an integer value $\{0, 1\}$ to indicate the UE association with the eNB *i*.

Table 3. Parameters used for evaluation.

	MeNB-AN	**SeNB-AN**	**BH Link**
Frequency band (GHz)	2	2.6	60
Available BW (MHz)	20 (BW_{PRB}= 0.18)	20 (BW_{PRB}= 0.18)	1000 (10 × 100 MHz)
Antenna gain (dBi) (G_{Tx}, G_{Rx})	<15	<15	36
$N_{ant_i}^{AN}$	4 (MIMO 4 × 4)	4 (MIMO 4 × 4)	1 for each active BH link
P_{0_i} (W)	130	6.8	3.9
P_{MAX}^{out} (W)	20	0.13	0.224
Δ_p	4.7	4.0	not used
Distance-dep. Path Loss	$128.1 + 37.6 \cdot \log_{10}(r)$ [68]	$140.7 + 36.7 \cdot \log_{10}(r)$ [68]	Equations (6)–(11) in [69]

As shown in Equation (3), the eNB has a statically minimum non-zero output power of the transceiver, $P_{0_i}^{AN}$, although there is no associated UE. Accordingly, switching off unused eNBs is critical to save energy. Table 3 shows experimental values for the aforementioned parameters in this study, such as $P_{max_i}^{AN}$ and $P_{0_i}^{AN}$.

4.1.2. BN Energy Consumption

The energy consumption of a BH link can be formulated similarly to the AN link: (i) static power ($P_{0_i}^{BH}$) of a transceiver for each backhaul link toward a next-hop eNB j, and (ii) dynamic power by the amount of aggregated user data rate that travels over that link:

$$e_{ij}^{BH} = P_{0_i}^{BH} + P_{out_i}^{BH_j} \tag{5}$$

where $P_{0_i}^{BH}$ represents the minimum non-zero static power of each BH transceiver at eNB i.

The dynamic power $P_{out_i}^{BH}$ of a mmWave backhaul link is derived by the multiplication of the band-wide transmission power $P_{t_i}^{BH_j}$ and bandwidth efficiency, as below:

$$P_{out_i}^{BH_j} = P_{t_i}^{BH_j} \cdot F_i^{BH_j} = P_{t_i}^{BH_j} \cdot \frac{\sum_{u \in \mathcal{U}} f_{ij}^u x_{ij}^u}{B_{ij}^{max}} \tag{6}$$

where B_{ij}^{max} is the maximum data rate for a backhaul link ij. The integer value x_{ij}^u indicates routing information if a data flow of a user u uses the backhaul link ij or not.

$$\begin{aligned} P_{t_i}^{BH_j} &= SNR + N_{th} + NF + PL + L_t \\ &\quad + L_r - G_t - G_r + L_m, \end{aligned} \tag{7}$$

where SNR is the signal-to-noise ratio satisfying B_{ij}^{max}, N_{th} stands for the thermal noise, NF stands for the noise figure and PL represents the free-space path loss. The parameters L_t and L_r represent the transmitter and receiver losses, respectively, while G_t and G_r are the transmitter/receiver antenna gains and L_m is the link margin.

The maximum transmitted power of a transceiver operating at frequency f_{BH} may be given by

$$P_{max_{i,BH}(dBm)} = EIRP_{max(dBm)} + T_{x_{loss}(dB)} - G_{Tx(dBi)}, \tag{8}$$

where $EIRP_{max}$ denotes the maximum equivalent isotropically radiated power, and $P_{max_i}^{BH}$ is configured as 224 mW according to specifications in [70], as shown in Table 3.

Total energy consumption of the BN is the sum of the energy consumption of the available backhaul links, as below.

$$e_i^{BN} = \sum_{j \in \mathcal{N}} e_{ij}^{BH} \tag{9}$$

As a consequence, the energy consumption of each eNB depends on user data flows and the static power consumption. Control message unicast or broadcast in the cell can consume extra energy in addition to the user traffic. In this study, we ignore energy consumption from the control overhead that is relatively less than the bearer. In the following section, therefore, we define several constraints to switch on or off the SeNBs based on the presence of the data flows.

4.2. Switch On and Off Model

We introduce two binary variables, s_i^{AN} and s_i^{BN}, that indicate whether the AN link and the BH link, respectively, is powered on or off at node i; that is:

$$s_i^{AN} = \begin{cases} 1 & \text{when AN at } i \text{ is powered on,} \quad \forall i \in \mathcal{N} \\ 0 & \text{when AN at } i \text{ is powered off,} \quad \forall i \in \mathcal{N} \end{cases} \tag{10}$$

$$s_i^{BN} = \begin{cases} 1 & \text{when all BH at } i \text{ are powered on,} \quad \forall i \in \mathcal{N} \\ 0 & \text{when all BH at } i \text{ are powered off,} \quad \forall i \in \mathcal{N} \end{cases} \tag{11}$$

The power status of the AN and BN, s_i^{AN} and s_i^{BN}, is decided by the use of access or backhaul links. Accordingly, switch variables for AN and BN are configured by the presence of data flows, as below:

$$s_i^{AN} \leq \sum_{u \in \mathcal{U}} f_{ij}^u \qquad \forall i \in \mathcal{S},\ \forall (i,j) \in \mathcal{L}_{\mathcal{AN}}{}^i \tag{12}$$

$$s_i^{BN} \leq \sum_{u \in \mathcal{U}} f_{ij}^u \qquad \forall i \in \mathcal{S},\ \forall (i,j) \in \mathcal{L}_{\mathcal{BH}}{}^i \tag{13}$$

For the multi-hop routing path of the user flows, a link (i,j) of power-off eNB i cannot be used as $x_{ij}^u = 0$:

$$\begin{aligned} \sum_{u \in \mathcal{U}} x_{ij}^u &\leq s_i^{AN} \cdot \varsigma, \quad \forall (i,j) \in \mathcal{L}_{\mathcal{AN}}{}^i \\ \sum_{u \in \mathcal{U}} x_{ij}^u &\leq s_i^{BN} \cdot \varsigma, \quad \forall (i,j) \in \mathcal{L}_{\mathcal{BH}}{}^i \end{aligned} \tag{14}$$

where ς is a big number (i.e., 10^8).

4.3. Multi-Hop Routing Model

In this section, routing constraints are given for user data flows in the mmWave backhaul mesh network. First, a user data flow should satisfy the flow conservation rule in Equation (15). Second, a user data flow travels along a single path rather than multiple paths in Equation (16); in this study, we only consider single connectivity rather than dual connectivity. Third, an UE therefore has to associate with only one eNB in Equation (17).

$$\sum_{j \in \mathcal{N}} f_{ij}^u - \sum_{j \in \mathcal{N}} f_{ji}^u = \begin{cases} R^u, & \text{if } i = \text{ source} \\ -R^u, & \text{if } i = \text{ sink} \\ 0 & \text{otherwise} \end{cases} \quad \forall u \in \mathcal{U}, \forall i \in \mathcal{N}, \tag{15}$$

where R^u represents the demanded data rate of each UE u.

$$\sum_{j\in\mathcal{N}} x_{ij}^u - \sum_{j\in\mathcal{N}} x_{ji}^u = \begin{cases} 1, & \text{if } i = \text{ source} \\ -1, & \text{if } i = \text{ sink} \\ 0 & \text{otherwise} \end{cases} \quad \forall u \in \mathcal{U}, \forall i \in \mathcal{N}, \tag{16}$$

where $x_{ij}^u = \{0, 1\}$ indicates the routing information of a user data flow, f^u.

$$\sum_{(iu)\in\mathcal{L}_{\mathcal{AN}}} x_{iu}^u = 1, \quad \forall u \in \mathcal{U} \tag{17}$$

4.4. Link Capacity and Scheduling Model

For capacity constraint, the data rate of each user flow and aggregated flows must be less than the access and backhaul link capacity. For instance, when more than one UE connects to the same eNB, they have to share the capacity on that access link.

Therefore, the AN capacity constraint is given as follows:

$$\sum_{u\in\mathcal{U}} \sum_{(i,u)\in\mathcal{L}_{\mathcal{AN}}} f_{iu}^u \leq C_i^{max}, \quad \forall i \in \mathcal{N}, \tag{18}$$

where C_i^{max} is the maximum capacity of eNB i as the access link capacity.

Additionally, the sum of the user flows on a given BH link is limited by the maximum capacity of the BH link:

$$\sum_{u\in\mathcal{U}} f_{ij}^u \leq c_{ij}, \quad \forall (i,j) \in \mathcal{L}_{\mathcal{BH}} \tag{19}$$

where $\mathcal{L}_{\mathcal{BH}}$ represents a set of BH links.

In the mmWave backhaul mesh network, we have to schedule transmissions among all links in the set of interference links, $(i, j) \in \mathcal{I}$. For duplex, first we adopt time division duplex (TDD), which is used to separate transmission and reception on a BH link (i.e., different time slots are assigned for the transmission from eNB i to j and for the transmission from eNB j to i). Similarly, time division multiplexing (TDM) is used to schedule transmissions among adjacent BH links. The following constraint ensures that the capacity of each BH link is shared among adjacent interferenced BH links:

$$\sum_{u\in\mathcal{U}} \left(\frac{f_{ij}^u x_{ij}}{b_{ij}} + \sum_{(kl)\in\mathcal{I}((ij))} \frac{f_{kl}^u x_{kl}}{b_{kl}} \right) \leq 1, \quad \forall i \text{ and } j \in \mathcal{N} \tag{20}$$

The flow rate on the link (i, j) can increase at the given link capacity as the interference is reduced by switching off SeNBs with the interfering BH links $(i, j) \in \mathcal{I}$.

4.5. Dual Objective Function

In this study, we have dual objectives, which are minimizing the total energy consumption of the HetNets while maximizing the sum of data rate R_u of each user u with the aforementioned constraints:

$$\min \omega_1 \sum_{i\in\mathcal{N}} e_i - \omega_2 \sum_{u\in\mathcal{U}} R^u \tag{21}$$

$$\text{s.t. Equations } (2) - (20)$$

where $\{\omega_1, \omega_2\}$ is a scaling vector that is used to impose weight for each objective; ω_1 and ω_2 are for energy consumption and throughput, respectively.

5. Deep Multi-Objective Reinforcement Learning in mmWave HetNet

In this section, we solve the optimization problem in Equation (21), which is not only non-convex, but contains dual objectives that are conflicting to each other. We introduce the PPO and PDOLS algorithms to effectively search for efficient solutions in the Pareto front of the dual objectives.

5.1. Proximal Policy Optimization

The TRPO is a stochastic policy-based optimization technique that can guarantee updates in the direction of increasing performance within a trust region. Schulman et al. [23] proposed a new policy optimization algorithm following the TRPO, called the PPO algorithm [24]. After then, several algorithms such as TD3 [71] and soft actor critic (SAC) [25] have been proposed, but the PPO is still a popular algorithm with some advantages of the TRPO. The PPO is easy to implement, using only first-order optimization, and is able to solve the data efficiency problem while achieving a similar performance as the complicated TRPO.

In the TRPO, updates are conducted by a policy that maximizes the objective function ("surrogate" objective) within a specific constraint as below,

$$\max_{\theta} \mathbf{E}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t \right] \tag{22}$$

$$\text{subject to } \mathbf{E}_t \left[\mathrm{KL}[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)] \right] \leq \delta \tag{23}$$

By applying the Kullback–Leibler divergence (KL) constraint between the old policy $\pi_{\theta_{old}}(a_t|s_t)$ and the current policy $\pi_\theta(a_t|s_t)$ in Equation (23), the TRPO can provide monotonical improvement to the $\pi_\theta(a_t|s_t)$ at each iteration and prevent excessive updates by limiting the range δ. However, it demands intensive computation for a rough solution that is infeasible to analyze. Instead, the constraint is relaxed by penalty with coefficient β in Equation (24), in which the surrogate objective forms a lower bound to guarantee the performance of the policy π.

$$\max_{\theta} \mathbf{E}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)} A_t - \beta \, \mathrm{KL}[\pi_{\theta_{old}}(\cdot|s_t), \pi_\theta(\cdot|s_t)] \right] \tag{24}$$

However, it is difficult to choose a constant value of β that performs well across various problems. For this, a new surrogate object function of the PPO is proposed to emulate monotonous improvement of the TRPO. The new surrogate objective function is presented in Equation (25),

$$L(\theta) = \mathbf{E}_t \left[\min(r_t(\theta) A_t, clip(r_t(\theta), 1-\epsilon, 1+\epsilon) A_t) \right] \tag{25}$$

Using the clip function, the PPO enables the surrogate objective function to avoid excessive policy updates while achieving similar performance to the TRPO. In addition, the PPO collects fixed-length T trajectory segments as a mini-batch and performs learning based on them repeatedly, which increases sample efficiency and learning stability.

For calculating A_t, a truncated version of generalized advantage estimation (GAE) is used,

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}, \tag{26}$$

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t). \tag{27}$$

Due to the high sample complexity (i.e., the number of training samples required for successfully learning) of our HetNet model that probably increases the number of necessary samples and their variance, we apply the truncated version of GAE, which provides stable and steady learning in the PPO algorithm [72]. GAE can enable monotonous increments in reward by reducing the sample variance through discount vector γ and λ like the $TD(\lambda)$.

5.2. MDP of mmWave-Backhaul HetNets

In this section, we define a MDP model (S, A, R^a, P^a_{ss}) for our multi-objective optimization problem in the HetNets.

- State S: the state in the HetNet MDP is denoted by a traffic matrix that represents traffic load $v_e = [0, 1]$ at access and backhaul links, which eventually determines throughput and energy consumption. In particular, we define a single representative state for all access links of a certain eNB instead of the individual state to reduce state information, since the AN energy consumption from transmission power, $P^{AN}_{out_i}$, is calculated by aggregated RBs of all associated users as shown in Equation (4). Accordingly, the vector size of the state space is $|\mathcal{L}_{\mathcal{BH}}| + |\mathcal{N}|$. We define the environment state, $s_t = \{v_1, v_2, \ldots, v_{|\mathcal{L}_{\mathcal{BH}}|}, v_{|\mathcal{L}_{\mathcal{BH}}|+1}, \ldots, v_{|\mathcal{L}_{\mathcal{BH}}|+|\mathcal{N}|}\}$, with v_e, as below:

$$v_e = \frac{1}{c_{ij}} \sum_{u \in U} f^u_{ij}, \qquad (i,j) \in \mathcal{L}_{\mathcal{BH}}, e = [1, |\mathcal{L}_{\mathcal{BH}}|] \tag{28}$$

$$= \frac{1}{C^{max}_i} \sum_{u \in U} f^u_{ij}, \qquad (i,j) \in \mathcal{L}_{\mathcal{AN}}, e = [|\mathcal{L}_{\mathcal{BH}}| + 1, |\mathcal{L}_{\mathcal{BH}}| + |\mathcal{N}|] \tag{29}$$

  where the index e of each link (i, j) is given by the environment at the beginning of the learning phase;
- Action A: the agent action is routing and association of user flows, which actually decides a set of x_{ij} binary variables, as discussed in Equations (16) and (17). However, such discrete action space grows exponentially by the number of the links, in which convergence of the learning algorithm is rarely guaranteed and large memory is required for computation. Instead, we consider a weight matrix ($a_t \in \mathbb{R}^{|\mathcal{L}|}$) of all links for all user flows, with which each flow finds a path using a link-state routing algorithm (e.g., the Dijikstra algorithm). Accordingly, the space complexity decreases from $O(2^{|\mathcal{L}|})$ to $O(|\mathcal{L}|)$. All actions for the links can be defined as below:

$$a_t = \{w_{ij} | w_{ij} \in \mathbb{R}, (i,j) \in \mathcal{L}\} \tag{30}$$

  Unfortunately, such a shortest path algorithm leads most of users to select a MeNB's AN link as a single-hop path; cumulative weights along a multi-hop path are mostly higher than for a single hop. This prevents the DRL algorithm from exploring actions of multi-hop routing that may offer reward gain by increasing user throughput, $\sum_{u \in \mathcal{U}} R^u$, more than the cost of energy consumption,$\sum_{i \in \mathcal{N}} e_i$.
  Therefore, we limit the number of user flows for the MeNB in the routing algorithm that admits the user flows to the MeNB only if the MeNB has available RBs, $\sum f^u_{ij} \leq C^{max}_i, i \in \mathcal{M}, j \in \mathcal{U}$. Otherwise, users find multi-hop paths through SeNBs in the algorithm;
- Reward R: the reward is given by the objective function of Equation (21). Thus, we change the minimization objective to maximization by multiplying Equation (21) by -1. For normalization, the sum rate of all UE flows and corresponding eNB energy consumption are divided by the sum of the maximum data rate and maximum energy consumption. Subsequently, the reward can be written in Equation (31) as

$$r_t = -\omega_1 \cdot r_e + \omega_2 \cdot r_d, \tag{31}$$

  where r_e and r_d represent $\sum_{i \in \mathcal{N}} \frac{e_i}{e_{max}}$ and $\frac{1}{|N|} \cdot \sum_{u \in \mathcal{U}} \frac{R^u}{d_u}$, respectively.

5.3. PPO-Based DRL for HetNet Optimization

The aforementioned MDP model of our HetNet optimization has continuous state and action spaces; thus, the PPO can effectively perform the exploration of solutions without the excessive updates in Equation (25). We implement the PPO-based DRL algorithm in Algorithm 1, which is based on the actor-critic architecture.

Algorithm 1 Proposed PPO Solution for mmWave HetNet

Input:

$\pi_\theta, V_\phi, \{\omega_1, \omega_2\}, Env$

Instruction:

1: **for** `iteration=1,2, ....,` **do**
2: **for** `iteration=1,2, ....,` T **do**
3: **for** `iteration=1,2, ....,` $|\mathcal{L}_{BH}| + |\mathcal{N}|$ **do**
4: $s_t = s_t \cup v_e$
5: **end for**
6: $a_t = \pi_{\theta_{old}}(s_t)$
7: $[r_e^t, r_d^t], s_{t+1} = Env(a_t)$
8: $r_t = -\omega_1 \cdot r_e + \omega_2 \cdot r_d$
9: $M = M \cup \{s_t, a_t, r_t, s_{t+1}\}$
10: $\hat{A}_t =$ compute advantage estimate from Equation (26)
11: **end for**
12: **for** `iteration=1,2, ....,` K **do**
13: update π_θ using Equation (32)
14: update V_ϕ using Equation (33)
15: **end for**
16: $\theta_{old} = \theta, \phi_{old} = \phi$
17: Drop M
18: **end for**

In the input of Algorithm 1, the actor network π_θ parametrized by θ provides a policy (a_t) according to the environmental state (s_t). Meanwhile the critic network presents the reward value $(V_\phi(s_t))$, which is parametrized by ϕ. At the beginning, the PPO collects total T trajectory tuples (S, A, R, S') (line 2–11), and subsequently, π_θ and V_ϕ are trained multiple K times with the T collected tuples (line 12–15). The parameters of π_θ and $V_\phi(s_t)$ are updated by Equations (32) and (33).

$$\theta = \arg\max_\theta \mathbf{E}_t[L(A_t, \theta_{old})] \tag{32}$$

where $L(A_t, \theta_{old})$ is derived by Equation (25) at the given old parameter θ_{old}.

$$\phi = \text{Smooth}_{L_1}(|V_\phi(s_t) - \widehat{V}_t^{GAE(\gamma,\lambda)}|) \tag{33}$$

where the $\widehat{V}_t^{GAE(\gamma,\lambda)}$ is a target value derived by Equation (26); that is, $\widehat{V}_t^{GAE(\gamma,\lambda)} = V_{\phi_{old}}(s_t) + \hat{A}_t$.

Since we implement both an actor network and a critic network, π_θ and $V_\phi(s_t)$ using multi-layer perceptrons(MLP), in the gradient update process, backward propagation is conducted; in this paper, we adopt Smooth_{L1} as an optimizer among Adagrad, Adam, Smooth_{L1}, etc. Although the surrogate objective function of the PPO in Equation (25) is applied only to π_θ, $V_\phi(s_t)$ is affected interactively within the actor-critic loop.Thereby, both policy and value can avoid excessive updates. The update process of the algorithm continues until the reward increases and converges to a certain level.

5.4. Multi-Objective Deep Reinforcement Learning

The PPO-based DRL algorithm can suffer from finding Pareto fronts in the multi-objective MDP (MOMDP) problem since it just learns a policy with a scalarized single objective which is unclear to evaluate each contribution of different objectives. As the reward of the MOMDP is a vector of n rewards of multi-objectives, $R(s_t, a_t) = r_t \in \mathcal{R}^n$ [47], for the reward scalarization, simple linearization such as $\mathcal{F}(V^\pi, \omega) = \omega \cdot V^\pi$ can be used (i.e., convex combination of the policy values, V^π), where V^π is a value vector for a policy, π, and ω is a weight vector for the importance of the objectives [48].

Therefore, we propose the PDOLS algorithm to find an optimal solution for the MOMDP problem. Figure 2 depicts how the PPO and the OLS cooperate for the multi-objective HetNet problem. The OLS part provides a framework of the outer loop to handle possible weight vectors, while the PPO part provides actor-critic networks to update the policy and value. The outer loop incrementally constructs the *convex coverage set* (CCS) that is an intermediate approximated coverage set, $\mathbb{S}$, by solving a series of single-objective MDPs scalarized by possible weight vectors, which eventually contains at least one optimal policy.

To reduce training efforts for all cases of weight vectors, the OLS manages corner weights that indicate break points in the piecewise linear CCS as a lower bound in addition to the $\mathbb{S}$. Thus, the OLS selects the weight vector for training only among the corner weights. When a new corner weight, ω', is discovered from the PPO learning, that is, $\exists v, \mathcal{F}(V^\pi, \omega') > v, v \in V_{\mathbb{S}}(\omega) = \{w \cdot V^\pi | V^\pi \in \mathbb{S}\}$, all scalarized values below $\mathcal{F}(V^\pi, \omega')$ are removed from $\mathbb{S}$. Afterwards, the OLS selects the next corner weight in a priority queue for learning, as shown in Figure 2. The detailed procedures of the PDOLS are described in Algorithm 2.

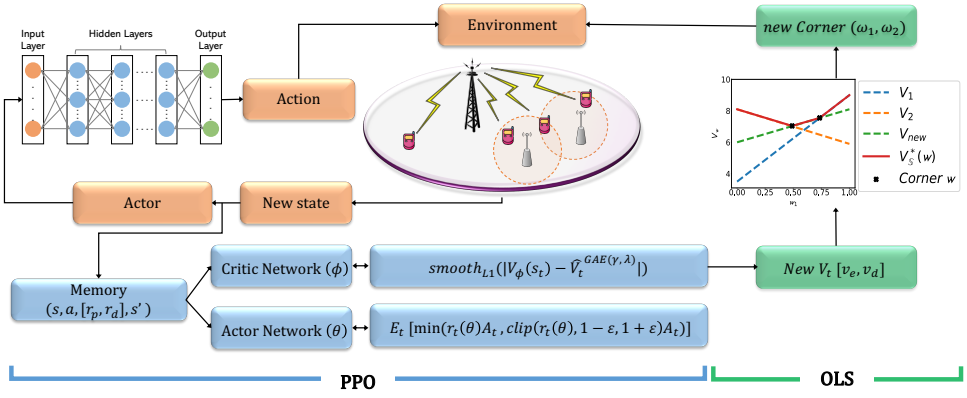


Figure 2. System architecture of the PPO-based deep OLS learning in mmWave HetNet.

The discovered corner weight ω_1 and ω_2 of energy consumption and throughput is used back for the PPO-based DRL to find a new lower bound of V^π and its π in the Algorithm 2 (line 5–18). At that time, the reward value r_e and r_d of energy consumption and throughput can affect the creation of a set of $V_{\mathbb{S}}^*(\omega)$. For instance, a new corner weight to be used for further learning and finding a new V^π is rarely found if the reward gap between two objectives is large. Therefore, we scale down the reward value instead of the original value from environment in order to increase the probability of finding the new corner weights (line 10). $\hat{A}_t$ is calculated through Equation (26) and $\hat{A}_t \in \mathbb{A}^2$ as $r_t \in \mathbb{R}^2$ (line 12–13). To reflect the corner weight from the OLS in $\hat{A}_t$, $\hat{A}_t$ is updated by multiplying $[A_e^t, A_d^t]$ and $[\omega_1, \omega_2]$ (line 13). When the convergence is achieved in the PPO learning process, the PPO sends a new $V_t[v_e, v_d]$ to the OLS (line 18).

Algorithm 2 PPO-Based Deep Optimistic Linear Support

Initialization:
$\mathbb{S}$ = partial CSS, which is composed of V_t obtained after the PPO learning.
$\mathbb{W}$ = corner weights, which is obtained from $\mathbb{S}$.
$\mathbb{Q}$ = priority queue of weights for the multi-objective, where the weights form a tuple along with their importance (i.e., $([\omega_1^t, \omega_2^t], I)$).

Instruction:

$\omega_t = \mathbb{Q}$.pop()
for `iteration=1,2, ....,` **do**
    for `iteration=1,2, ....,` T **do**
        $a_t = \pi_{\theta_{old}}(s_t)$
        $[r_e^t, r_d^t], s_{t+1} = Env(a_t)$
        Reduce scaling of $[r_e^t, r_d^t]$
        $M = M \cup \{s_t, a_t, [r_e^t, r_d^t], s_{t+1}\}$
        $[A_e^t, A_d^t]$ = compute advantage estimate from Equation (26)
        $\hat{A}_t = \hat{A}_t \cup \{[A_e^t, A_d^t] \times [\omega_1^t, \omega_2^t]\}$
    end for
    Optimize surrogate L and wrt θ from $\hat{A}_t$, with K epochs
    Optimize V_ϕ and wrt ϕ from $\widehat{V}_t^{GAE(\gamma,\lambda)}$, with K epochs
    $\theta_{old} = \theta, \phi_{old} = \phi$
end for when convergence
$V_t = V_\phi(s)$
$\mathbb{W} = \mathbb{W} \cup \omega_t$
if $\omega_t \cdot V_t > \sum_{U \in \mathbb{S}} \omega_t \cdot U$ **then**
    $\mathbb{S}$ = remove obsolete V_{del} due to new V_t
    ω_c = new corner weight from $\mathbb{S}$
    $\mathbb{S} = \mathbb{S} \cup V_t$
    $\mathbb{Q}$ = remove obsolete ω_{del} due to new ω_c
    for `iteration=1,2, ....,` ω_c **do**
        if estimate improvement of $(\omega', \mathbb{W}, \mathbb{S}) > \tau$ **then**
            $\mathbb{Q} = \mathbb{Q} \cup \omega'$
        end if
    end for
end if
if $\mathbb{Q}$ is not empty **then**
    go back to line 1
end if

The priority queue of the weights, $\mathbb{Q}$, is initially configured with extreme weights (i.e., $[0, 1], [1, 0]$) and updated whenever a new corner weight is found. The priority is determined according to the distance between $\mathcal{F}(V^\pi, \omega')$ of the new corner weight ω' and a line made by values of two adjacent corner weights on both sides of the new corner

weight. In other words, the priority is proportional to the degree of convexity downward in $V^*_{\mathbb{S}}(\omega)$.

The OLS removes obsolete V_{del} and ω_{del} when creating a new $V^*_{\mathbb{S}}(\omega)$ (line 22, 25). Depending on the improvement of the new corner weight, the OLS decides whether to add it to the $\mathbb{Q}$ by comparing to a threshold τ (line 26–28). We set the τ to 0 to train aggressively for all discovered corner weights to find optimal values. Finally, the PPO and OLS stop processing if no new corner weight is found and $\mathbb{Q}$ is empty (line 32-33).

6. Experiment

In this section, we evaluate the performance in terms of energy saving and user throughput, comparing algorithms proposed in the previous section. We establish an experimental environment with 1 MeNB and 25 SeNBs that form a backhaul mesh network as depicted in Figure 3, where the mmWave BH links (i.e., gray dashed lines in Figure 3) connect the SeNBs to each other or to the MeNB for Internet access. There are only 4 SeNBs reachable to the MeNB, which thus limits the sum rate of all data flows below the sum of their BH link capacity. Therefore, we assume that each UE, u, demands a maximum 14 Mbps data rate (d_u) in this experiment with the 100 UEs and last mile 4 SeNBs since those bottleneck BH links (i.e., the purple dot line in Figure 3) allow 14 Mbps per UE. To support a greater UE data rate, we can increase the BH link bandwidth or place more SeNBs reachable to the MeNB gateway.

A total of 100 UEs are randomly dropped over the MeNB and SeNB coverage area, where the SeNBs are apart by 100 meters and their cell coverage is more than 80 meters. Accordingly, the UEs have more than one SeNB to associate with, in addition to the universal MeNB, depending on their location. Both the MeNB and SeNBs provide microwave link access, denoted by AN links in Figure 3. The access and BH link is configured as in Table 3 for our experiment. In our study, the training and model update are performed interactively with the network simulator environment based on parameters specified in 3GPP standard and related works [68,69].

We build actor-critic networks using a DNN with 2 hidden layers (64×64 perceptrons) of a fully-connected neural network to estimate the policy and value, respectively. The actor network for policy receives the input of the state field and returns the action field as output as defined in Section 5.2. On the other hand, the critic network for value is designed differently according to the PPO and PDOLS algorithm. Both algorithms receive the same input for the state field, but the PPO-based critic returns only one value, while the PDOLS-based critic returns two values of the dual objectives. Detailed parameters for the DRL are introduced in Table 4. For this experiment, we used the pyTorch library on a Linux 20.04 server equipped with Intel CPU i7-9700KF, GPU GeForce RTX 2080 and 32 GB RAM.

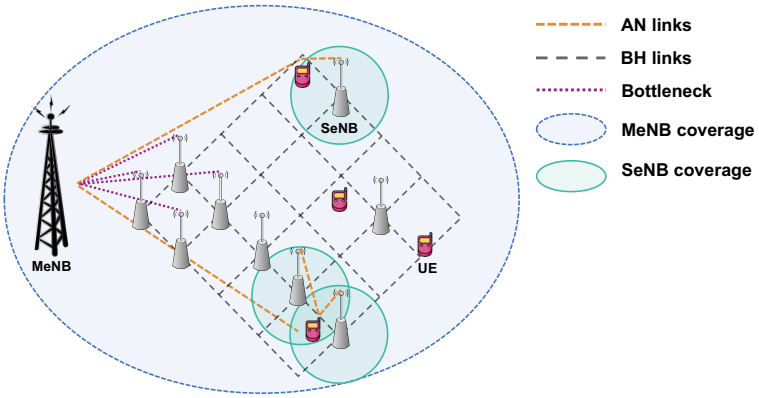
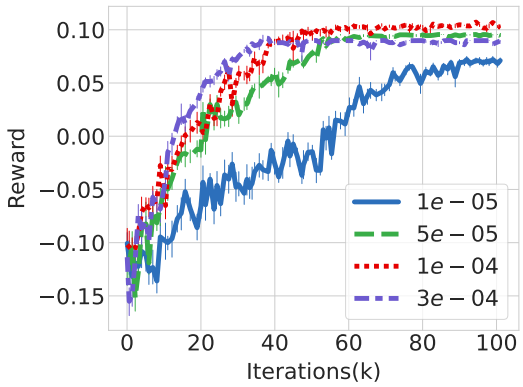


Figure 3. Experimental HetNet topology.

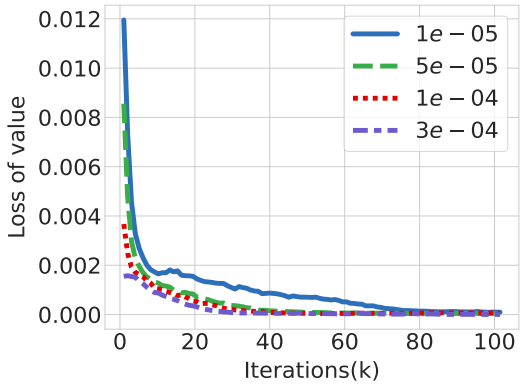
Table 4. Training hyperparameters.

Parameter	Value	Parameter	Value
γ	0.8	λ	0.8
Trajectory size	1024	Batch size	32
K epoch	10	Clipping range ϵ	0.2
Learning rate of actor	1×10^{-4}	Learning rate of critic	1×10^{-4}
Network initialization	HE	Optimization method	Smooth_{L1}

First, we evaluate the performance of the PPO-based DRL algorithm in the HetNet environment in terms of learning speed and convergence. For this, we configure the weight vector of energy consumption and data rate as $\omega_1 = 0.5$ and $\omega_2 = 0.5$, respectively, and the UE demand rate as 14 Mbps. Figure 4a shows the performance with varying learning rates from 1×10^{-5} to 3×10^{-4}. The PPO algorithm shows good convergence of reward as training iterations continue, regardless of learning rate. The reward increases exponentially during the initial training iterations and becomes saturated after 40 K training iterations. The higher learning rate accelerates the reward convergence, but it skips over the better local minimum and is trapped in another; when the learning rate increases from 1×10^{-4} to 3×10^{-4}, the converged reward decreases from 0.104 to 0.0899. The loss for the value and policy can be seen in Figures 4b,c, respectively. The loss of value and policy decreases drastically as the training iterations continue. Policy learning can avoid excessive learning owing to clipping of the PPO, which leads the policy loss to be comparable regardless of the learning rate. Additionally, the value loss follows the policy loss through the actor-critic interactions.

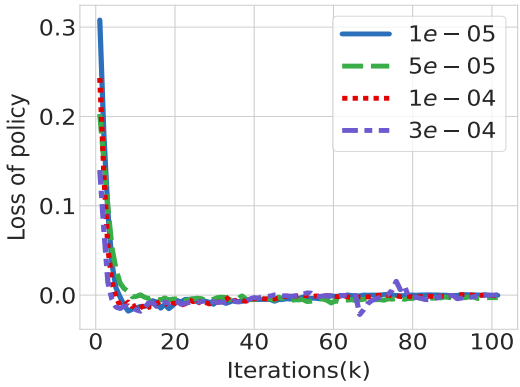


(**a**)



(**b**)

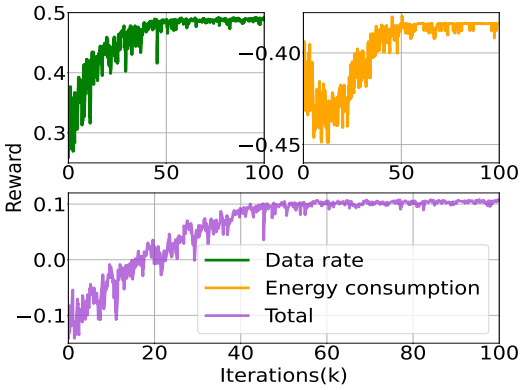
Figure 4. *Cont.*



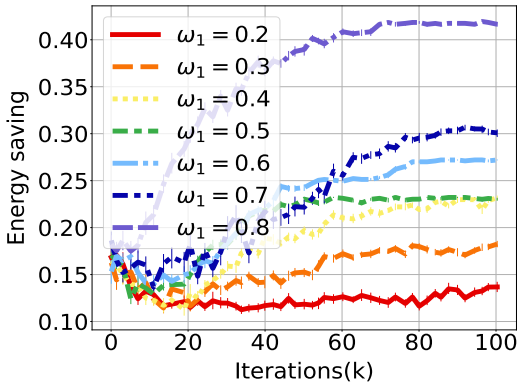
(**c**)

Figure 4. Performance evaluation of PPO according to learning rate. (**a**) Reward convergence. (**b**) Value loss. (**c**) Policy loss.

Figure 5 shows evaluations on learning performance with varying reward weights (ω_1, ω_2). For this experiment, we configure the learning rate as 1×10^{-4}, which shows the fastest convergence with the highest reward. In Figure 5a, rewards from energy consumption and user throughput converge at 50 K training iterations with reward weight ($\omega_1 = 0.5, \omega_2 = 0.5$).

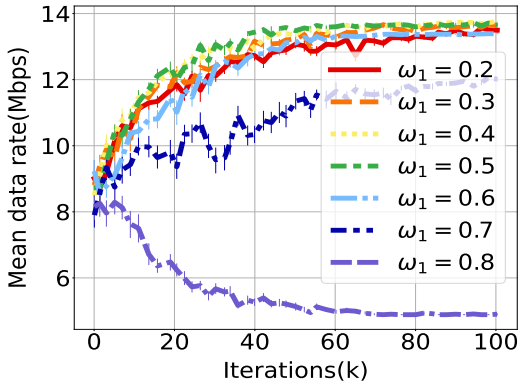


(**a**)



(**b**)

Figure 5. *Cont.*



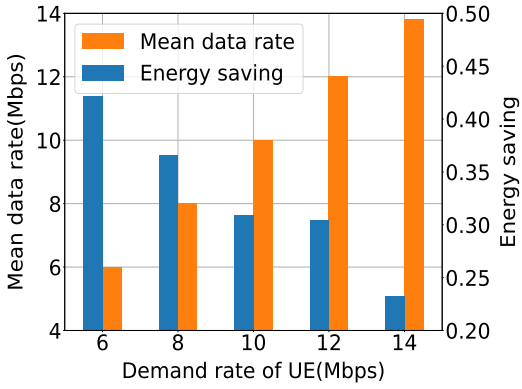
(**c**)

Figure 5. Performance evaluation of PPO according to reward weight ($\omega_2 = 1 - \omega_1$). (**a**) Scalarized reward. (**b**) Energy savings. (**c**) Average data rate per UE.

Figures 5b,c show that the energy saving (i.e., 1-consumed energy/maximum energy) and mean data rate converge at different iterations according to the reward weight; the reward convergence is achieved at an average of 80 K training iterations, about 21.5 min on our server for each weight value. To find the optimal solutions, iterative learning for all possible weight vectors is needed. Therefore, the computation delay depends on the granularity of the weight values to explore; this experiment demands a total of 80 K · 7 iterations.

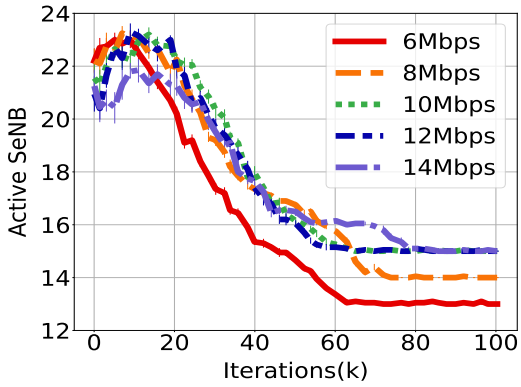
System performance varies with ω_1 of the energy consumption from 0.2 to 0.8 and ω_2 of the UE's data rate, $1 - \omega_1$. When ω_1 is set to 0.8, the maximum energy saving is achieved by 0.419, while the UE's data rate is only 4.89 Mbps as a minimum value, because of their trade-off relationship. Contrarily, the minimum energy saving, 0.134, allows the maximum data rate, 13.7 Mbps, with $\omega_1 = 0.2$. Consequently, the optimal weight for maximum reward is found to be $\omega_1 = 0.6$ and $\omega_2 = 0.4$, which results in an energy savings of 0.272 and a UE data rate of 13.39 Mbps.

Next, we evaluate the PDOLS algorithm to find the optimal value and weight in a HetNet environment with a varying demand rate and number of UEs. In Figure 6a, the mean data rate satisfies most of all demand rates except for 14 Mbps: 6, 8, 10, 12, and 13.39 Mbps. The energy savings of the HetNet is inversely proportional to the demand rate: 0.42, 0.37, 0.31, 0.30, and 0.23. For these values, the ω_1 of the optimal weight is 0.79, 0.72, 0.65, 0.64 and 0.57, with respect to each data rate.



(**a**)

Figure 6. *Cont*.



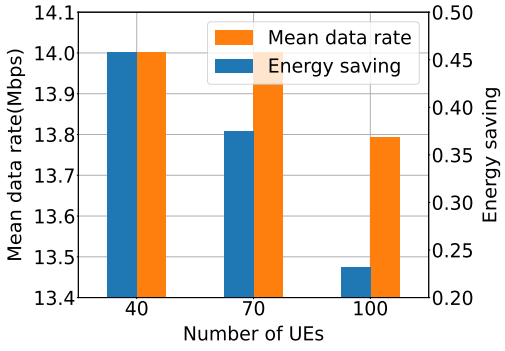
(**b**)

Figure 6. Performance evaluation of PDOLS according to UE's demand rate. (**a**) Average data rate per UE. (**b**) Number of active SeNBs.

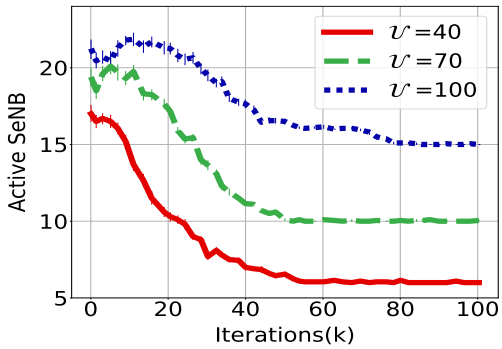
Figure 6b shows the change of the active SeNBs during the learning procedure. Most of the 25 SeNBs are turned on at the beginning of learning, but after 80K iterations, almost 10–12 SeNBs are switched off according to the UE's demand rate. For the higher demand rate, more SeNBs are active to support the user traffic. Although the number of active SeNBs is the same for 10, 12, and 14 Mbps, energy consumption increases, especially for the 14 Mbps in Figure 6a, as power consumption of the active links increases proportionally by user traffic.

We evaluate the performance of the PDOLS again with different numbers of UEs such as 40, 70 and 100, where the demand data rate is configured to be 14 Mbps. Figure 7a shows that both energy savings and the sum of the data rate increase as the number of UEs decreases. Accordingly, the user demand rate is mostly satisfied, except for 100 UEs. The energy saving is 0.46, 0.38, and 0.2, respectively, for each number of UEs. The corresponding active SeNBs are 6, 10, and 15, as shown in Figure 7b. Here, ω_1 of the optimal weight is found to be 0.8, 0.66, and 0.57 for each case. For 40 UEs, the number of active SeNBs is around 18 initially and decreases to up to 6 SeNBs, as data flows of many UEs use the same multi-hop paths provided by the active SeNBs. Otherwise, isolated UEs that have no path through the SeNBs directly access to the MeNB. Comparing the result of 100 UEs with 6 Mbps, we can conjecture that a higher number of UEs induces network-wide deployment, which consumes more RBs of the MeNB and transmission power for a smaller number of serving UEs.

Figure 8a compares the performance of the proposed algorithms discussed in Section 5, where the number of UEs and the demand rate are configured as 100 and 14 Mbps. A heuristic algorithm leads the UEs to associate with a less-loaded SeNB and use the shortest path to the MeNB gateway, which performs worse with energy savings of 0.16 and a data rate of 9.14 Mbps than others. Meanwhile, the PPO and PDOLS show comparable results of 0.27, 13.39 Mbps for the PPO and 0.23, 13.79 Mbps for the PDOLS, where the optimal weight for the PPO is selected manually after iterative executions with different weight vectors, while the PDOLS algorithm automatically searches for the optimal weight values. The PDOLS-SR outperforms other algorithms with 0.27 and 13.79 Mbps when the reward is scaled by 1/5.

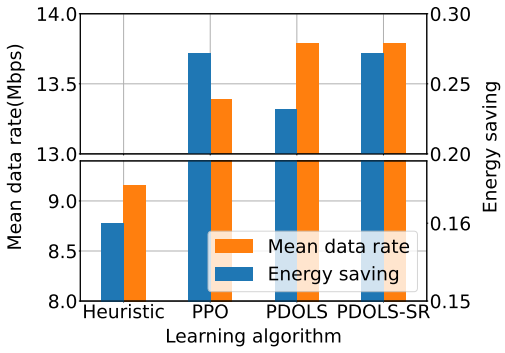


(**a**)

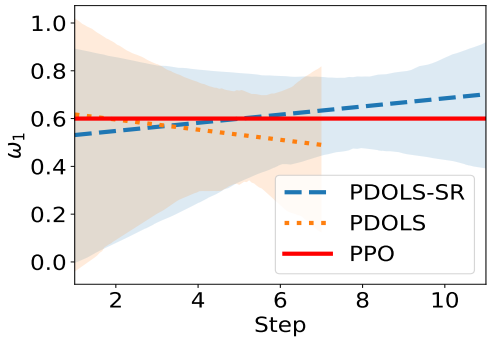


(**b**)

Figure 7. Performance evaluation of PDOLS according to the number of distributed UEs. (**a**) Average data rate per UE. (**b**) Number of active SeNBs.



(**a**)



(**b**)

Figure 8. Performance comparison of proposed algorithms. (**a**) Average UE data rate and energy consumption. (**b**) Reward weight vector exploration.

Figure 8b shows the variation of corner weight in the OLS framework of the PDOLS. In our experiment, the PDOLS-SR conducts the training process 11 times (11 steps in the figure) to find the optimal weight, while the PDOLS does this only 7 times (7 steps). The PDOLS-SR can scavenge and explore more corner weights to find a near-optimal weight close to the PPO weight, 0.6 (the red solid line). The optimal ω_1 of the PDOLS-SR is 0.5872, while the ω_1 of the PDOLS is 0.5683. Further adjustment for downscaling of the reward, such as 1/10 or 1/15, only increases training time without notable performance enhancement.

7. Conclusions

In this paper, we solve a multi-objective optimization problem of throughput maximization and energy consumption minimization in a HetNet with a mmWave-backhaul mesh. For this, we implement a PPO-based DRL algorithm based on actor-critic architecture. However, the conventional PPO algorithm has limitations in its ability to cope with the multi-objective problem. Therefore, we propose PDOLS, which allows the PPO algorithm to interoperate with OLS as an outer loop to search for an optimal weight vector for the dual objectives. Experimental results show that the PPO-based DRL algorithm converges successfully with increasing rewards as training is iterated. Additionally, the learned solution of energy saving and user throughput is comparable to the CPLEX result. PDOLS can find a feasible weight vector for the dual objectives which is similar to the optimal weight that is identified manually using all possible combinations of the weight values.

Author Contributions: W.K. conceived and designed the main idea and wrote the paper; K.R. developed algorithm and performed the experiments. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Technology Development Program (no.G21S290100202) funded by the Ministry of SMEs and Startups (MSS, Korea). This research was supported by the Gachon University Research Fund 202008460005.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rappaport, T.S.; Sun, S.; Mayzus, R.; Zhao, H.; Azar, Y.; Wang, K.; Wong, G.N.; Schulz, J.K.; Samimi, M.; Gutierrez, F. Millimeter wave mobile communications for 5G cellular: It will work! *Access IEEE* **2013**, *1*, 335–349. [CrossRef]
2. Sun, S.; MacCartney, G.R.; Samimi, M.K.; Nie, S.; Rappaport, T.S. Millimeter wave multi-beam antenna combining for 5G cellular link improvement in New York City. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 5468–5473.
3. Maccartney, G.R.; Rappaport, T.S. 73 GHz millimeter wave propagation measurements for outdoor urban mobile and backhaul communications in New York City. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014; pp. 4862–4867.
4. Dehos, C.; González, J.L.; De Domenico, A.; Ktenas, D.; Dussopt, L. Millimeter-wave access and backhauling: The solution to the exponential data traffic increase in 5G mobile communications systems? *Commun. Mag. IEEE* **2014**, *52*, 88–95. [CrossRef]
5. Wang, P.; Li, Y.; Song, L.; Vucetic, B. Multi-gigabit millimeter wave wireless communications for 5G: From fixed access to cellular networks. *Commun. Mag. IEEE* **2015**, *53*, 168–178. [CrossRef]
6. Taori, R.; Sridharan, A. Point-to-multipoint in-band mmwave backhaul for 5G networks. *IEEE Commun. Mag.* **2015**, *53*, 195–201. [CrossRef]
7. Zhu, Y.; Niu, Y.; Li, J.;Wu, D.O.; Li, Y.; Jin, D. QoS-Aware Scheduling for Small Cell Millimeter Wave Mesh Backhaul. Available online: https://ieeexplore.ieee.org/document/7511065 (accessed on 10 October 2021).
8. Nakamura, M.; Tran, G.K.; Sakaguchi, K. Interference Management for Millimeter-Wave Mesh Backhaul Networks. Available online: https://ieeexplore.ieee.org/document/8651725 (accessed on 10 October 2021).
9. Zola, E.; Kassler, A.J.; Kim, W. Joint User Association and Energy Aware Routing for Green Small Cell Mmwave Backhaul Networks. Available online: https://ieeexplore.ieee.org/document/7925706 (accessed on 10 October 2021).
10. Jaber, M.; Imran, M.A.; Tafazolli, R.; Tukmanov, A. 5G backhaul challenges and emerging research directions: A survey. *IEEE Access* **2016**, *4*, 1743–1766. [CrossRef]

11. Correia, L.M.; Zeller, D.; Blume, O.; Ferling, D.; Jading, Y.; Gódor, I.; Auer, G.; Van Der Perre, L. Challenges and enabling technologies for energy aware mobile radio networks. *IEEE Commun. Mag.* **2010**, *48*, 66–72. [CrossRef]
12. Ashraf, I.; Boccardi, F.; Ho, L. Sleep mode techniques for small cell deployments. *IEEE Commun. Mag.* **2011**, *49*, 72–79. [CrossRef]
13. Soh, Y.S.; Quek, T.Q.; Kountouris, M.; Shin, H. Energy efficient heterogeneous cellular networks. *IEEE J. Sel. Areas Commun.* **2013**, *31*, 840–850. [CrossRef]
14. Suárez, L.; Nuaymi, L.; Bonnin, J.M. Energy-efficient BS switching-off and cell topology management for macro/femto environments. *Comput. Netw.* **2015**, *78*, 182–201. [CrossRef]
15. Liu, C.; Pan, Z.; Liu, N.; You, X. A Novel Energy Saving Strategy for LTE HetNet. Available online: https://ieeexplore.ieee.org/document/6096845 (accessed on 10 October 2021).
16. Bhaumik, S.; Narlikar, G.; Chattopadhyay, S.; Kanugovi, S. Breathe to Stay Cool: ADJUSTING Cell Sizes to Reduce Energy Consumption. Available online: https://dl.acm.org/doi/10.1145/1851290.1851300 (accessed on 10 October 2021).
17. Chen, L.; Yu, F.R.; Ji, H.; Rong, B.; Li, X.; Leung, V.C. Green full-duplex self-backhaul and energy harvesting small cell networks with massive MIMO. *IEEE J. Sel. Areas Commun.* **2016**, *34*, 3709–3724. [CrossRef]
18. Mesodiakaki, A.; Adelantado, F.; Alonso, L.; Di Renzo, M.; Verikoukis, C. Energy-and spectrum-efficient user association in millimeter-wave backhaul small-cell networks. *IEEE Trans. Veh. Technol.* **2016**, *66*, 1810–1821. [CrossRef]
19. Hao, W.; Zeng, M.; Chu, Z.; Yang, S.; Sun, G. Energy-efficient resource allocation for mmWave massive MIMO HetNets with wireless backhaul. *IEEE Access* **2017**, *6*, 2457–2471. [CrossRef]
20. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2018.
21. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic Policy Gradient Algorithms. Available online: http://proceedings.mlr.press/v32/silver14.pdf (accessed on 10 October 2021).
22. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
23. Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; Moritz, P. Trust Region Policy Optimization. Available online: https://arxiv.org/abs/1502.05477 (accessed on 10 October 2021).
24. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
25. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. Available online: https://arxiv.org/abs/1801.01290 (accessed on 10 October 2021).
26. Wang, S.; Liu, H.; Gomes, P.H.; Krishnamachari, B. Deep reinforcement learning for dynamic multichannel access in wireless networks. *IEEE Trans. Cogn. Commun. Netw.* **2018**, *4*, 257–265. [CrossRef]
27. Zhong, C.; Lu, Z.; Gursoy, M.C.; Velipasalar, S. Actor-Critic Deep Reinforcement Learning for Dynamic Multichannel Access. Available online: https://ieeexplore.ieee.org/document/8646405 (accessed on 10 October 2021).
28. Zhong, C.; Lu, Z.; Gursoy, M.C.; Velipasalar, S. A deep actor-critic reinforcement learning framework for dynamic multichannel access. *IEEE Trans. Cogn. Commun. Netw.* **2019**, *5*, 1125–1139. [CrossRef]
29. Naparstek, O.; Cohen, K. Deep Multi-User Reinforcement Learning for Dynamic Spectrum Access in Multichannel Wireless Networks. Available online: https://arxiv.org/abs/1704.02613 (accessed on 10 October 2021).
30. Naparstek, O.; Cohen, K. Deep multi-user reinforcement learning for distributed dynamic spectrum access. *IEEE Trans. Wirel. Commun.* **2018**, *18*, 310–323. [CrossRef]
31. Li, Y.; Zhang, W.; Wang, C.X.; Sun, J.; Liu, Y. Deep reinforcement learning for dynamic spectrum sensing and aggregation in multi-channel wireless networks. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 464–475. [CrossRef]
32. Liu, S.; Wu, J.; He, J. Dynamic Multichannel Sensing in Cognitive Radio: Hierarchical Reinforcement Learning. *IEEE Access* **2021**, *9*, 25473–25481. [CrossRef]
33. He, Y.; Yu, F.R.; Zhao, N.; Yin, H.; Boukerche, A. Deep Reinforcement Learning (DRL)-Based Resource Management in Softwaredefined and Virtualized Vehicular Ad Hoc Networks. Available online: https://www.semanticscholar.org/paper/Deep-Reinforcement-Learning-(DRL)-based-Resource-in-He-Yu/e1d5360a49ee5269298a54c49d171661ffc245d6 (accessed on 10 October 2021).
34. Shi, W.; Li, J.; Wu, H.; Zhou, C.; Cheng, N.; Shen, X. Drone-cell trajectory planning and resource allocation for highly mobile networks: A hierarchical DRL approach. *IEEE Internet Things J.* **2020**, *8*, 9800–9813. [CrossRef]
35. Wang, X.; Zhang, Y.; Shen, R.; Xu, Y.; Zheng, F.C. DRL-based energy-efficient resource allocation frameworks for uplink NOMA systems. *IEEE Internet Things J.* **2020**, *7*, 7279–7294. [CrossRef]
36. Ahsan, W.; Yi, W.; Qin, Z.; Liu, Y.; Nallanathan, A. Resource allocation in uplink NOMA-IoT networks: A reinforcement-learning approach. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 5083–5098. [CrossRef]
37. Rahimi, A.M.; Ziaeddini, A.; Gonglee, S. A novel approach to efficient resource allocation in load-balanced cellular networks using hierarchical DRL. *J. Ambient. Intell. Humaniz. Comput.* **2021**, 1–15. [CrossRef]
38. Ali, R.; Shahin, N.; Zikria, Y.B.; Kim, B.S.; Kim, S.W. Deep reinforcement learning paradigm for performance optimization of channel observation–based MAC protocols in dense WLANs. *IEEE Access* **2018**, *7*, 3500–3511. [CrossRef]
39. Yu, Y.; Wang, T.; Liew, S.C. Deep-reinforcement learning multiple access for heterogeneous wireless networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1277–1290. [CrossRef]

40. Al-Tam, F.; Correia, N.; Rodriguez, J. Learn to Schedule (LEASCH): A Deep reinforcement learning approach for radio resource scheduling in the 5G MAC layer. *IEEE Access* **2020**, *8*, 108088–108101. [CrossRef]
41. Nisioti, E.; Thomos, N. Robust coordinated reinforcement learning for MAC design in sensor networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2211–2224. [CrossRef]
42. Liu, C.H.; Chen, Z.; Tang, J.; Xu, J.; Piao, C. Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2059–2070. [CrossRef]
43. Liu, J.; Zhao, B.; Xin, Q.; Su, J.; Ou, W. DRL-ER: An Intelligent Energy-aware Routing Protocol with Guaranteed Delay Bounds in Satellite Mega-constellations. *IEEE Trans. Netw. Sci. Eng.* **2020**. [CrossRef]
44. El Amine, A.; Dini, P.; Nuaymi, L. Reinforcement Learning for Delay-Constrained Energy-Aware Small Cells with Multi-Sleeping Control. Available online: https://ieeexplore.ieee.org/document/9145431 (accessed on 10 October 2021).
45. Asuhaimi, F.A.; Bu, S.; Klaine, P.V.; Imran, M.A. Channel access and power control for energy-efficient delay-aware heterogeneous cellular networks for smart grid communications using deep reinforcement learning. *IEEE Access* **2019**, *7*, 133474–133484. [CrossRef]
46. Hsieh, C.K.; Chan, K.L.; Chien, F.T. Energy-efficient power allocation and user association in heterogeneous networks with deep reinforcement learning. *Appl. Sci.* **2021**, *11*, 4135. [CrossRef]
47. Roijers, D.M.; Vamplew, P.; Whiteson, S.; Dazeley, R. A survey of multi-objective sequential decision-making. *J. Artif. Intell. Res.* **2013**, *48*, 67–113. [CrossRef]
48. Mossalam, H.; Assael, Y.M.; Roijers, D.M.; Whiteson, S. Multi-objective deep reinforcement learning. *arXiv* **2016**, arXiv:1610.02707.
49. Dong, P.; Zhang, H.; Li, G.Y.; Gaspar, I.S.; NaderiAlizadeh, N. Deep CNN-based channel estimation for mmWave massive MIMO systems. *IEEE J. Sel. Top. Signal Process.* **2019**, *13*, 989–1000. [CrossRef]
50. Lin, B.; Wang, X.; Yuan, W.; Wu, N. A novel OFDM autoencoder featuring CNN-based channel estimation for internet of vessels. *IEEE Internet Things J.* **2020**, *7*, 7601–7611. [CrossRef]
51. Jiang, P.; Wen, C.K.; Jin, S.; Li, G.Y. Dual CNN based Channel Estimation for MIMO-OFDM Systems. *IEEE Trans. Commun.* **2021**, *69*, 5859–5872. [CrossRef]
52. Zheng, S.; Qi, P.; Chen, S.; Yang, X. Fusion methods for CNN-based automatic modulation classification. *IEEE Access* **2019**, *7*, 66496–66504. [CrossRef]
53. Huynh-The, T.; Hua, C.H.; Pham, Q.V.; Kim, D.S. MCNet: An efficient CNN architecture for robust automatic modulation classification. *IEEE Commun. Lett.* **2020**, *24*, 811–815. [CrossRef]
54. Hermawan, A.P.; Ginanjar, R.R.; Kim, D.S.; Lee, J.M. CNN-based automatic modulation classification for beyond 5G communications. *IEEE Commun. Lett.* **2020**, *24*, 1038–1041. [CrossRef]
55. Vinayakumar, R.; Soman, K.; Poornachandran, P. Applying convolutional neural network for network intrusion detection. In Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 13–16 September 2017; pp. 1222–1228.
56. Kim, J.; Kim, J.; Kim, H.; Shim, M.; Choi, E. CNN-based network intrusion detection against denial-of-service attacks. *Electronics* **2020**, *9*, 916. [CrossRef]
57. Riyaz, B.; Ganapathy, S. A deep learning approach for effective intrusion detection in wireless networks using CNN. *Soft Comput.* **2020**, *24*, 17265–17278. [CrossRef]
58. Azizjon, M.; Jumabek, A.; Kim, W. 1D CNN Based Network Intrusion Detection with Normalization on Imbalanced Data. Available online: https://ieeexplore.ieee.org/document/9064976 (accessed on 10 October 2021).
59. Zhao, N.; Liang, Y.C.; Niyato, D.; Pei, Y.; Wu, M.; Jiang, Y. Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 5141–5152. [CrossRef]
60. Zhang, Q.; Liang, Y.C.; Poor, H.V. Intelligent user association for symbiotic radio networks using deep reinforcement learning. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 4535–4548. [CrossRef]
61. Ding, H.; Zhao, F.; Tian, J.; Li, D.; Zhang, H. A deep reinforcement learning for user association and power control in heterogeneous networks. *Ad Hoc Netw.* **2020**, *102*, 102069. [CrossRef]
62. Du, Z.; Deng, Y.; Guo, W.; Nallanathan, A.; Wu, Q. Green Deep Reinforcement Learning for Radio Resource Management: Architecture, Algorithm Compression, and Challenges. *IEEE Veh. Technol. Mag.* **2020**, *16*, 29–39. [CrossRef]
63. Dai, Y.; Zhang, K.; Maharjan, S.; Zhang, Y. Edge intelligence for energy-efficient computation offloading and resource allocation in 5G beyond. *IEEE Trans. Veh. Technol.* **2020**, *69*, 12175–12186. [CrossRef]
64. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
65. Sutton, R.S.; Precup, D.; Singh, S. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artif. Intell.* **1999**, *112*, 181–211. [CrossRef]
66. Sutton, R.S.; McAllester, D.A.; Singh, S.P.; Mansour, Y. Policy Gradient Methods for Reinforcement Learning with Function Approximation. Available online: https://proceedings.neurips.cc/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0f-Paper.pdf (accessed on 10 October 2021).
67. Auer, G.; Giannini, V.; Desset, C.; Godor, I.; Skillermark, P.; Olsson, M.; Imran, M.A.; Sabella, D.; Gonzalez, M.J.; Blume, O.; et al. How much energy is needed to run a wireless network? *IEEE Wirel. Commun.* **2011**, *18*, 40–49. [CrossRef]

68. 36.814. Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for E-UTRA Physical Layer Aspects. Available online: https://www.scirp.org/(S(czeh2tfqyw2orz553k1w0r45))/reference/ReferencesPapers.aspx?ReferenceID=998750 (accessed on 10 October 2021).
69. Mesodiakaki, A.; Adelantado, F.; Antonopoulos, A.; Kartsakli, E.; Alonso, L.; Verikoukis, C. Energy Impact of Outdoor Small Cell Backhaul in Green Heterogeneous Networks. Available online: https://ieeexplore.ieee.org/document/7033196?arnumber=7033196 (accessed on 10 October 2021).
70. gTSC0020, gAPZ0039, gRSC0016, gRSC0015, gTSC0023, gAPZ0042. Technical Report, Gotmic. 2017. Available online: www.gotmic.se (accessed on 10 October 2021).
71. Fujimoto, S.; Hoof, H.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. Available online: https://arxiv.org/abs/1802.09477 (accessed on 10 October 2021).
72. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv* **2015**, arXiv:1506.02438.