# Reinforcement Learning Approach for Resource Allocation in 5G HetNets

Fivos Allagiotis[1], Christos Bouras[1], Vasileios Kokkinos[1], Apostolos Gkamas[2], Philippos Pouyioutas[3]
[1]Computer Engineering and Informatics Department, University of Patras, Patras, Greece
[2]University Ecclesiastical, Academy of Vella, Ioannina, Greece
[3]Computer Science Department, University of Nicosia, Nicosia, Cyprus
st1056636@ceid.upatras.gr, bouras@upatras.gr, kokkinos@cti.gr, gkamas@aeavellas.gr, pouyioutas.p@unic.ac.cy

***Abstract*— Heterogeneous Networks (HetNets) have been hailed as a critical technology for 5G communications, allowing for the rapid expansion of mobile traffic. HetNets can increase network capacity and serve additional users by installing small cells inside macrocells. However, resource allocation for such networks becomes more challenging than for conventional cellular networks due to interference between small-cells and macrocells, making it more difficult to provide quality of service. Deep Reinforcement Learning (DRL) has opened the door for applications in resource allocation for 5G HetNets, because of recent breakthroughs in the field. We present a unique resource allocation technique based on DRL that may be used to both small and macro cells. According to the resource allocation process, an autonomous "agent", in our case a cell, makes judgments to determine the appropriate BS to assign to a user, and the optimal number of Resource Blocks that should be allocated, while not needing or waiting for any information.**

***Keywords— 5G, Resource Allocation, HetNets, Reinforcement Learning, Deep Q-Learning***

## I. INTRODUCTION

Individuals and organizations may have various quality of service (QoS) needs, and 5G networks are required to include a wide range of services. The rapid growth of data traffic has highlighted the necessity to increase the capacity of future increased networks. Consequently, 5G networks need to be more adaptable and expandable. The 3rd Generation Partnership Project (3GPP) offered Heterogeneous Networks (HetNets) as a solution. In HetNets, macro cells are used for coverage. Pico and micro cells are utilized in crowded regions to increase capacity. The installation of these small cells is an important aspect of the HetNet strategy since it allows for a lot of flexibility where they may be placed. In addition, HetNets can increase network capacity and resource efficiency by reusing time, space, and frequency resources [1]. HetNets are widely regarded as a potent method in next-generation cellular networks, with substantial research [2]. The key issues HetNets face are load balancing and interference coordination, both of which are related to Resource Allocation (RA). One common issue is that when all subscribers choose to connect to the macro cells based on the highest receiving signal power, the traffic would be distributed unevenly. On the other hand, the power and bandwidth of the macro cells will be underused if all users connect to the nearest picocell [3].

Deep Reinforcement Learning (DRL) has been developed as a viable answer to the problem of RA. DRL can deal with high-dimensional state spaces and achieving improved results. This paper's primary purpose can be summarized as follows: Creating a simulated environment that resembles a traditional 5G network, where an agent will be installed to learn how to best allocate the network's resources. Reinforcement Learning (RL) is made up of three key concepts: state, action, and reward. The state in our example will represent the network's current state in terms of throughput, energy, and QoS. An agent can assign users to certain Base Stations (BSs) or more Resource Blocks (RBs) to remote users. In our case, the RBs will be distributed to satisfy the needs of all users and maximize the total throughput inside the network. Simultaneously, the algorithm will be running with specific energy constraints, to enhance the energy efficiency of the network. When an agent performs an action in a state, it is rewarded. When the reward is positive, it corresponds to what we normally think of as a reward. When the reward is negative, it is equivalent to a "punishment". What we described above, will be implemented using a common DRL algorithm, known as Deep Q-learning.

This concept has been examined before by many researchers, but not to the extent that we propose. Indicatively, work [4] proposed decentralized Q-learning based processes to achieve interference coordination. Each picocell is a self-contained entity that chooses how to expand the range of the cell and the transmission power in each channel based on the detected SINR state. We believe that a single entity that controls the whole operation of the network might provide better results, as any race conditions will be discarded, and lower the complexity of system. In [5], the authors suggested a distributed RL solution for cellular network power control. Each cell acts as an agent, selecting its own transmission power based on the average Reference Signal Received Power (RSRP) and Signal to Interference Noise Ratio (SINR) of the cell's users. We on the other hand, will develop a model that not only assigns the transmission power of the cell, but also manages other forms of resources, such as user association and RBs.

In [6], a Multi-Agent Double Deep-Q Network technique was developed to establish a distributed optimal strategy. Each user is treated as an agent, and the BSs that will be connected, as well as the transmission channels that will be used, are chosen based on the QoS of all users. Treating, all users as a single agent will increase the complexity of the overall system, resulting in a larger computational cost, and increased demands in computer resources. The technique in [7] fulfills the Macrocell Base Station (MBS) users' QoS while attempting to maximize the network's total capacity. However, QoS and fairness between femtocell BS (FBS) subscribers are not considered. The authors of [8] adopt a round robin strategy to boost the throughput of cell-edge users and, simultaneously maintain equity amongst MBS and FBS clients. In [9] cooperative Q-learning was utilized to optimize the cumulative capacity of FBS users whilst preserving the macro's user capacity around a set limit. Despite this, the QoS of FBS users is not considered in [8] or [9].

The rest of the paper is organized as follows. In Section II, we describe the system model we used. The proposed mechanism is described in detail in Section III, where an example is also provided. Further, in Section IV, we evaluate our proposal after using it in a simulated environment and producing tables and graphs, with the results. Finally, the conclusions and future steps are provided in Section V.

## II. System Model

Because the linked Smallcell Base Stations (SBS) will be closer than the standard MBS, the introduction of heterogeneous SBS can boost network capacity and improve network coverage. The proposed configuration of future HetNets is illustrated in Fig. 1. SBSs are densely distributed inside the coverage zone of regular MBSs, which in this scenario use a hexagonal configuration. We assume that the same telecommunications company deploys and maintains SBSs, and that these SBSs have logical links to the MBS. SBSs can unload traffic from the MBS, and the MBS makes intelligent decisions about user scheduling and resource distribution. We suppose that there are $K$ SBSs and that the set $K = \{0, 1, 2, \dots, K\}$ of BSs serves a set $M = \{0, 1, 2, \dots, M\}$ of users. The sequential decision-making process takes place in discrete time slots, each with a fixed duration $T_s$. The goal of user scheduling is to assign each user to a single BS.

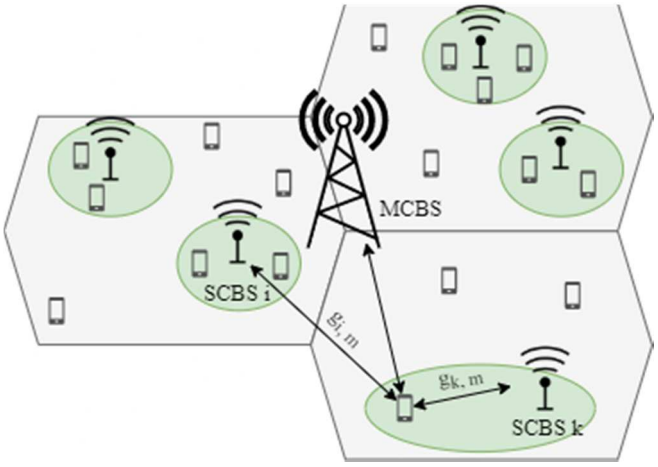


Fig. 1. Typical HetNets scenario.

5G BSs have limited resources, significantly in relation of frequency bandwidth [10] The Physical Resource Block (PRB) is the smallest allocation unit for a 5G BS, consisting of 12 frequency subcarriers with a $2^{\mu} \times 15$ kHz bandwidth, where $\mu \in \{0,1,2,3,4\}$ is the numerology parameter. The maximum allowable bandwidth on each BS and its numerology, as described by 5G NR standards [10] determine the number of PRBs accessible on each BS. As a result, the user scheduling decision for BS $k$ at time slot $t$ is $u_k(t) \in \{0, 1, \dots, M\}$ which indicates BS $k \in K$ serves $u_k(t)$ users at time slot $t$, while $\sum_{k=0}^{K} u_k(t) = M$. It further indicates that BS $k$ provides user $m$ with a certain number $W_p$ of Physical Resource Blocks (PRBs). Using $w_{pk}^{m}$ we characterize the amount of PRBs that the BS $k$ provides to user $m$.

The goal of resource allocation is to find the best transmission power and amount of PRBs for scheduled users. The receiving power, or RSRP, is the transmission power measured between the UE $m$ and the BS $k$. It is calculated as such:

$$RSRP_{m,k} = P_k + G_k - L_k - L_{m,k} \quad (1)$$

where $P_k$ represents the BS's output power, $G_k$ represents the antenna gain, $L_k$ represents the feeder loss, and $L_{m,k}$ represents the computed path loss. The BS's output power is defined as follows:

$$P_k = \frac{1000 \cdot BS_{power}}{\frac{BS_{PRB}}{10 \cdot 2^{\mu}} \cdot BS_{subcarriers}} \quad (2)$$

$BS_{PRB}$ represents the total amount of the minimum distribution unit, for a 5G BS, the PRB. The number of PRBs available in the BS depends on the total bandwidth available and its numerology, as defined by the 5G NR standards in [11].

The BS $k$ computes the SINR after a UE $m$ makes a connection request/update to determine the bitrate each of its resource blocks can offer. It can calculate the number of PRBs to be allocated for the UE connection using this value.

$$SINR_{m,k} = \frac{RSRP_{m,k}}{I+N} \quad (3)$$

where $RSRP_{m,k}$ is the strength of the incoming signal of interest, $I$ is the strength of the other (interfering) signals in the network, and $N$ is a noise factor, which can be constant or random. The interference is computed according to the other BSs visible by the UE, their RSRP, their utilization ratio and the utilization ratio of the BS involved in the connection. Moreover, the data rate that can be transmitted by assigning a PRB to the UE $m$ using the Shannon formula, and is as follows:

$$r_{m,k} = 2^{-\mu} 10^{-3} \cdot B_{PRB} \log_2(1 + SINR_{m,k}) \quad (4)$$

where $B_{PRB}$ is the bandwidth of an individual PRB and can be calculated as:

$$B_{PRB} = 12 \cdot 2^{\mu} 15\ kHz \quad (5)$$

Now, given a requested bit rate $b_k^m$ from UE $m$, it is possible to calculate the number of PRBs and the bitrate to be allocated by BS $k$ to satisfy the request, using the formulas below:

$$n_{m,\mathrm{k}}^{PRB} = \left\lceil (b_k^m / r_{m,\mathrm{k}}) \right\rceil \quad (6)$$

$$ab_k^m = n_{m,\mathrm{k}}^{PRB} \cdot r_{m,k} \quad (7)$$

The simulator also provides various metrics for the BSs, the UEs and the network. Namely, the power consumption of each BS, the total power consumption of the network and the QoS of each UE, amongst others. The total consumption of every BS $k$ is measured by multiplying the amount of energy that a BS exerts to transfer a single PRB, with the total number of PRB that it allocates to the users. The formula is shown below:

$$E_k = \sum_{m \in M} n_{m,k}^{PRB} \cdot \frac{P_k}{12 \cdot BS_{PRB}} \quad (8)$$

It is then evident, that to retrieve the total energy consumption of the network, we must simply retrieve the sum of the power consumption of each BS, thus concluding to:

$$E_{net} = \sum_{k \in K} E_k \quad (9)$$

As QoS, our simulator, outputs two versions of the term, one for each individual user and one for the entirety of the network. For each UE, we define the QoS as shown below:

$$QoS_{UE}^{m} = \frac{ab_m^k}{b_m^k} \quad (10)$$

while for the network, the QoS mainly reflects the percentage of the total number of users, that are being serviced, by the network, and as such, it is calculated as:

$$QoS_{net} = \frac{UE_{connected}}{UE_{total}} \quad (11)$$

The metrics mentioned above, will be used to compare the proposed algorithm, in Section IV.

## III. Problem Formulation and Proposed Solution

The Markov Decision Process (MDP), Q-learning and the components of our Deep Q-learning approach will be explained in this section. Following that, the unification of the suggested methodology and small cell cooperation are discussed.

### A. Markov Decision Process

$\{S, A, T, R, \Sigma, \gamma\}$ is the tuple's definition of an MDP, where S and $A$ are the finite state (continuous or discrete) and action set, respectively, $T$ is the function $T : S \times A \times S \to [0, 1]$, that calculates the probability of a transition occurring with $T(s, a, s')$ denoting the likelihood that the next action is $s'$ when the current state is $s$ and the chosen action is $\alpha$, and with $\sum_{s' \in S} T(s, a, s') = 1$, $R$ is the reward function for a single step $R: S \times A \times S \to \mathbb{R}$, $\Sigma$ is the original allocation of states and $\gamma \in (0,1)$ ensures that future incentives are weighed against present rewards by a discount factor $\gamma$. Because not all actions are available in every stage, the collection of actions may be state-dependent. $A(s) \subseteq A$ represents the collection of actions accessible at a given state s ∈ S.

For each state, a deterministic policy $\pi: S \to A$ chooses one action. Let $\Pi$ denote the collection of policies $\pi$ that are viable for everybody, such that $\pi(s) \in A(s)$ for all $s \in S$. The state-value function represents the estimated discounted reward achieved by beginning from state $s$ and following policy $\pi$ subsequently, and is defined as:

$$V_\pi(s) = E_\pi(\sum_t \gamma^t R(s_t, a_t, s_{t+1}) | s_0 = s) \quad (12)$$

where $E_\pi$ is the policy's $\pi$ anticipated value, $s_t$ and $a_t$ indicate the state and action at time $t$. Likewise, the state-action value function:

$$Q_\pi(s, a) = E_\pi(\sum_t \gamma^t R(s_t, a_t, s_{t+1}) | s_0 = s\,, a_0 = \alpha) \quad (13)$$

reflects the expected discounted reward when beginning from a state $s$ and performing action $a \in A(s)$ by following the policy $\pi$. The MDP is solved by determining the best policy $\pi^*$ that maximizes the expected cumulative discounted reward. RL techniques, such as Q-learning, on the other hand, seek to predict the ideal state-action-value function $Q_{\pi^*}$ based on the controller's experience interacting with the environment.

### B. Q-Learning

The usual Q-learning update rule is:

$$Q(x_t, a_t) \leftarrow (1 - a) Q(x_t, a_t) + a \max_a (E[R_t + \gamma Q(x_{t+1}, a)]) \quad (14)$$

where $r_t = R(s_t, a_t, s_{t+1})$ denotes the measured reward at time $t$ and $a_t > 0$ is the learning rate, which is subject to the requirements $\sum_{t=1}^{\infty} a_t = \infty$ and $\sum_{t=1}^{\infty} a_t^2 < \infty$ in order to ensure convergence. Q-learning is specified in procedural form in Algorithm 1 [12].

**Algorithm 1** Q-Learning algorithm

1: Initialize $Q(x_t, a_t)$ arbitrarily
2: **for all** episodes **do**
3: Initialize $x_t$
4: **for all** steps of episode **do**
5: Choose $a_t$ from set of actions
6: Act $a_t$, observe $R_t, x_{t+1}$
7: Perform Eq. 13
8: $x_t \leftarrow x_{t+1}$
9: **end for**
10: **end for**

The parameter $\epsilon_t \in [0,1]$ in so-called $\epsilon$-greedy strategies control the balance between exploration and exploitation: The agent selects a random action with probability $\epsilon_t$ at any time $t$, while selecting the action that maximizes the state-action-value function with probability $1 - \epsilon$.

It is important to note that the Q function is only adjusted in traditional RL techniques for the examined state-action pairings. To obtain a thorough approximation of the optimum Q function, each state-action combination must be visited at least once. This means that the state space and action space A must be limited and discrete, and RL algorithms suffer from the so-called dimensionality problem as their dimensions grow. The Deep Q-Network (DQN) technique was presented in [13] as a deep learning alternative for function approximation-based Q-learning to overcome these concerns. DQN uses a deep neural network that can estimate high-dimensional functions with a low-dimensional form to approximate the Q function. Despite including some technical solutions to address neural network constraints, such as the target network and memory buffers, the training process for the neural network is described in [13] and it is theoretically the same as in conventional Q-learning, with (13) replaced by the neural network training process.

### C. Deep Q-Learning

We must first define our state space. As previously stated, each BS is identified by the quantity of PRBs available for allocation, $W_p$. The state of the network should collect details about: 1) the load of the PRBs over the different BSs, and 2) the coverage quality that the BSs provides to the UE to make it possible for the agent to make effective decisions on the allocation of the PRBs on existing, or even new incoming requests from varying UEs. Let us represent the amount of PRBs allocated at any one time $t$ to maintain the services that have been assigned (to support the minimum bitrate level of each service), as $NPRB_k(t)$. It is then derived that:

$$NPRB_k(t) = \sum_{k\in K} \sum_{m\in M} w_{pk}^m \tag{15}$$

Then, let $L_p^k(t)$ be the load level of a BS $k$ at time slot $t$, which is defined as the ratio of allocated PRBS to the amount of available PRBS. We have by definition:

$$L_p^k(t) = \frac{NPRB_k(t)}{W_p} \tag{16}$$

Thus, the observed state $S_m$ for a UE $m$ seeking a service is described by the following two components: 1) The load level of each BS, that the UE receives a signal from, and 2) The value of the reference signals received power (RSRP) that the UE gets from every BS in its vicinity, as calculated by the UE. Thus, the state space is defined as:

$$S_m(t) = \left\{ s_m^k(t) = \left\{ L_p^k(t), RSRP_m^k \right\} \right\} \tag{17}$$

Additionally, we must also define the action space for our agent. Whenever a new connection request is received by the network controller, one of two things can happen: 1) The controller processes the request and assigns it to one specific BS. 2) Due to a lack of resources, the connection is refused since no BSs can handle it. We now define the action set. Let $A_m$ be a vector of size $K$, and of the same dimension as $s_m^k \in S_m$. The vector is initialized at the $MIN\_RSRP$ value which is equal to $-140$. Each position holds the RSRP value that the user receives from the BS in the vicinity of UE $m$. More specifically, the position $k$ of the vector $A_m(s)$ contains the measured RSRP that the UE $m$ receives from BS $k$. If the new connection request is granted, the largest element in $A_m$ reflects the BS from which UE $m$ receives the best signal. As a result, a request service may be assigned on BS $k$ in each state $s_m^k$ if and only if $s_m^k \in S_m$, meaning that by assigning the new request to the BS, the newly created state remains in $S_m$. So, the action set available in a state $s \in S$ is then defined as:

$$A_m(s) = \{RSRP_1, \dots, RSRP_k\} \tag{18}$$

Based on the above, regarding the action space, the unusual situations of that $a_i = 0$ indicates that the connection request must be refused due to a shortage of network resources, since no BS can assign the incoming request at the minimum needed bitrate.

Lastly, we must describe the reward function. To achieve this, we must first define $b_m^k$ $b_{pk}$ as the bitrate UE $m$ receives from BS $k$ and the extra bitrate that the BS $k$ may offer to the connection using a portion of its remaining resources, respectively. Furthermore, we define $b_{req}$ as the bitrate that UE $m$ requests. The QoS profile connected with the connection is directly tied to this amount. The reward function must then account for four scenarios:

1. The connection is assigned to a BS and the UE receives the proper amount of resources.
2. The connection is assigned to a BS and the UE receives an excess number of resources.
3. The connection is assigned to a BS with limited resources, thus the UE lacks the resources.
4. The request for a connection is denied (i.e., no BS receives the connection).

To account for all four scenarios, the agent's reward function $R_t(s_t, a_t, s_{t+1})$ while assigning a service $i$ to a BS $k$ will be specified as

$$R_t = \begin{cases} \frac{b_m^k}{L_p^k(t)}, & b_m^k > b_{req} \\ {b_m^k}^2 b_{pk} L_p^k(t), & b_{pk} > b_m^k \\ {b_m^k}^2 b_{pk}, & b_{req} > b_m^k \\ -p, & A_m(s) = \{\mathbf{0}\} \end{cases} \tag{19}$$

The negative reward is a penalty, represented by $-p$, provided to the agent if the allocation is denied. If the allocation is denied, then the action vector will be filled with zeros. It is intended to help the agent learn that if a connection is refused, then the agent must look for neighboring cells to (at least) connect the UE to a BS. The rest of the features are as follows:

TABLE I. DEEP Q-LEARNER PARAMETERS

| ***Parameters*** | ***Value*** |
|---|---|
| Learning rate | 0.7 |
| Discount factor | 0.618 |
| Initialized $\epsilon$ | 1 |
| $\epsilon_{max}$ | 1 |
| $\epsilon_{min}$ | 0.01 |
| $\epsilon$ reduction rate | 0.01 |
| Number of episodes | 500 |

## IV. PERFORMANCE EVALUATION

The sample simulation scenario is summarized in the table below. The scenario was created by a simulator developed in the context of our research laboratory's operations. The goal is to test the algorithm's capacity to associate/distribute users across three distinct radio access technologies (Macro, Micro, and Pico), while optimizing downlink (DL) UE data rate and decreasing BS power transmission. Also, the system should keep track of how many PRBs a user receives. The number of PRBs assigned to a user should be just enough to ensure that the service requested by the user can be provided adequately. The simulator was built using the [14] and [15] 3GPP standards. The channel model is described in [14] for a variety of frequencies and circumstances. The paper describes many scenarios, antenna models, path loss, and attenuation that may be chosen depending on the environment. Finally, [15] presents the transmitter and receiver characteristics that make up our simulated network.

TABLE II. SIMULATION PARAMETERS

| *Parameters* | *Values* | | |
|---|---|---|---|
| | *Macro* | *Micro* | *Pico* |
| Carrier Frequency (MHz) | 2100 | 2400 | 2600 |
| Bandwidth (MHz) | 5 | 5 | 5 |
| Maximum DL power (W) | 1 | 0.25 | 0.1 |
| Maximum BS power (W) | 20 | 2 | 1 |
| Antenna gain (dB) | 16 | 5 | 5 |
| Path loss (dB) | 3 | 2 | 2 |
| UE antenna gain (dB) | 0 | 0 | 0 |

The layout of the Macrocells is a hexagonal grid with a cell radius 200 meters, while the Microcells, Picocells and the UEs are uniformly distributed inside the radius of the Macrocell. Besides that, noise is added, following a normal distribution, to the signals that are being transmitted from the BSs to the UEs. Lastly, as Key Performance Indicators (KPIs) we define the QoS, the networks overall bitrate and the energy efficiency of the network. The number of users varies from 5000 to 35000 users.

As mentioned, the agent learns according to the reward he receives. The reward is calculated using the function, we defined in Section III. The simulator is being run each time. In every simulated instance the users are assigned, and the number of PRBs each user receives is adjusted. For each simulated network, the reward the agent will receive is recalculated. The agent aims to increase the reward he receives, thus at every step he learns to make the optimal assignment of users to BSs, but also the optimal allocation of PRBs to the assigned users. The training was performed, in a period of 500 episodes. The reward the agent receives for each episode, is presented, in the Figure below:

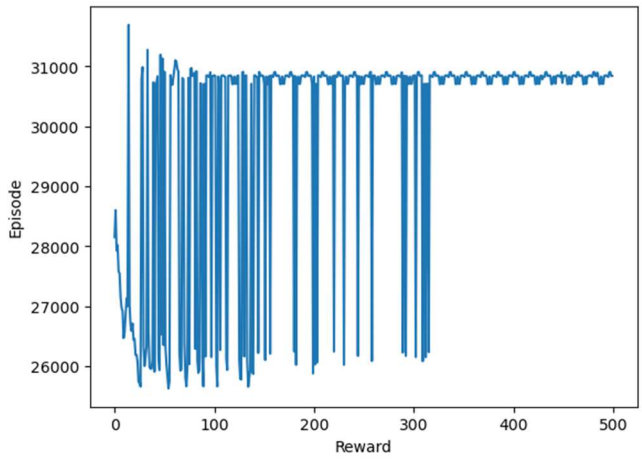


Fig. 2. The rewards received over the training episodes.

As we can see, the reward amount that the agent receives, fluctuates over the number of episodes, going from 25500 to 31500. These fluctuations can be interpreted as the agent exploring the environment. When the rewards are low, the agent is making random assignments. This pattern continues, up until the first 300 episodes. From there we can see that the reward settles at around 31000, which is an indication that the agent no longer explores its environment; rather, it makes calculated and optimal assignments of users to BS as well as optimal allocations of PRBs to the users.

A typical resource allocation approach is used to compare our Deep Q-Learning algorithm. This method solely takes into account the RSRP received from BSs in the users' immediate area. It follows the BS with the highest RSRP and, as a result, the strongest signal. The user is then assigned to the specified BS. The number of PRBs assigned to a user is determined by the user's service requirements as well as the RSRP that the user receives. To compare the two algorithms, we consider three metrics: (1) the total throughput, (2) the power consumption, and (3) the QoS of the network.

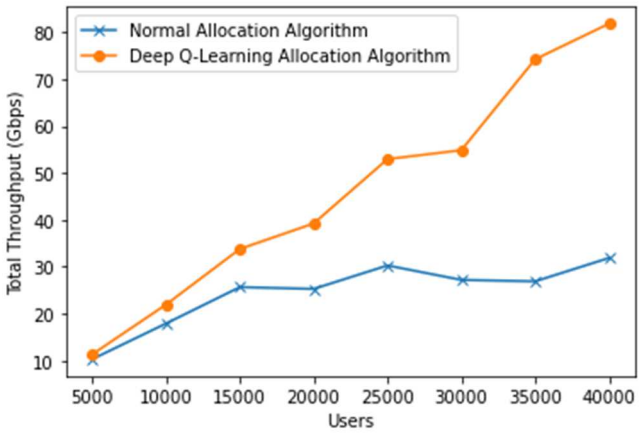


Fig. 3. The total throughput of the network, measured in Gbps.

It is evident, from Figure 3, that our Deep Q-Learning algorithm greatly outperforms the normal allocation algorithm, which is based on the received RSRP values from the BS. The Deep Q-Learning algorithm far exceeds the other in all cases, and after a certain number of users, it begins to double and even nearly triple the total throughput of the simulated network. This occurs since our mechanism ensures that nearly all the users are connected to the network (as it will be shown in Figure 5). Thus, all the users receive a certain amount of bitrate, rather than have only a fraction of the users receiving unnecessary amounts of PRBs.

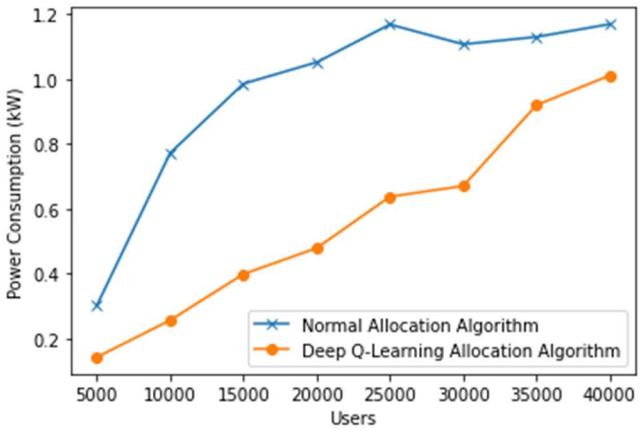


Fig. 4. The total power consumption of the network, measured in kW.

Figure 4 displays the total power consumption inside our simulated network. In terms of energy consumption, our mechanism ensures that the energy is distributed at acceptable levels, despite the fact that there are more users connected to the network. Energy consumption is calculated from the sum of the power consumption of all BSs. It is a function of the number of PRBs that are allocated throughout the network. To find how much energy each BS consumes, we calculate how

much power it requires to transmit a single PRB and multiply it by the number of PRBs that a specific BS assigns to the users. Because our mechanism ensures that the number of PRBs is exactly what each user requires, it keeps the energy consumption lower than a classic algorithm that assigns more PRBs than those that correspond to each user.

Finally, Figure 5 shows how the QoS of the network changes as the number of users increases. The first thing we can discern is how much the classic algorithm is affected by the increase of users. With each increase, the QoS produced by this algorithm decreases from 10% -20%. On the contrary, our mechanism keeps the QoS stable, in all but one case. The results show that all users are connected to the network. In the case where the QoS drops slightly, by 4%, it is the case where it could be addressed by adding more SBS, i.e., exceeding the capabilities of the mechanism.

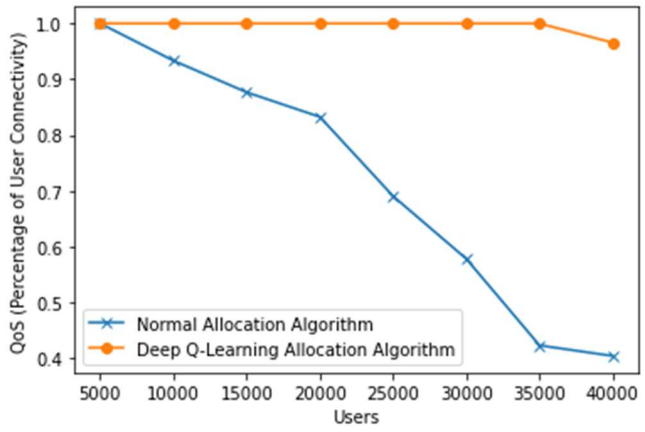


Fig. 5. The QoS of the network, as the number of users increases.

## V. Conclusions and Future Work

The research presented a deep reinforcement learning-based network controller for better resource allocation in 5G HetNets. By expressing the problem of user association and PRB distribution as a Markov decision process, the suggested controller was compared to a typical benchmark method. We created an open-source network simulator for testing purposes that properly captures the network resource utilization of several radio technologies, mimicking a 5G HetNet. In comparison to the other technique examined, the suggested controller increased network performance by raising the connection-flow acceptance rate and offering better resource management. Furthermore, the network's power consumption was lowered while the network's overall throughput was enhanced.

To summarize, Machine Learning, especially Reinforcement Learning, is introduced into the system, resulting in an intelligible and long-lasting process that solves a variety of issues, including the one described in this study. The findings indicate that, given the right input parameters, the suggested model efficiently adapts to the ever-changing environment while also delivering a stable mobile network system.

The technique is implemented with three principles in mind: how to boost overall throughput, how to conserve the most amount of energy, and how to keep the QoS at an acceptable level. The findings are encouraging, indicating that the network's overall throughput may be significantly increased. In addition, it demonstrates that increased throughput does not always imply higher energy consumption, and that a method that effectively assigns PRBs may help save a significant amount of money on the BS side of the network. All the above benefits are obtained while ensuring that the QoS does not fall below specified thresholds; if it does, it is usually a sign that more BS should be added to the network.

A future implementation of this technique might add more data and variables to train the model, increasing the amount of reward received by the agent and allowing for even better resource allocation and user association.

## VI. Acknowledgment

This research has been co-financed by the Hellenic Foundation for Research & Innovation (H.F.R.I) through the H.F.R.I.'s Research Projects to Support Faculty Members & Researchers (project code: 02440).

## References

[1] N. Saquib, E. Hossain and D. I. Kim, "Fractional frequency reuse for interference management in LTE-advanced hetnets," in IEEE Wireless Communications, vol. 20, no. 2, pp. 113-122, April 2013.

[2] D. Liu et al., "User Association in 5G Networks: A Survey and an Outlook", in IEEE Communications Surveys & Tutorials, vol. 18, no. 2, pp. 1018-1044, Secondquarter 2016.

[3] J. G. Andrews, S. Singh, Q. Ye, X. Lin and H. S. Dhillon, "An overview of load balancing in hetnets: old myths and open problems," in IEEE Wireless Communications, vol. 21, no. 2, pp. 18-25, April 2014.

[4] M. Simsek, M. Bennis and A. Czylwik, "Dynamic Inter-Cell Interference Coordination in HetNets: A reinforcement learning approach," 2012 IEEE Global Communications Conference (GLOBECOM), Anaheim, CA, 2012, pp. 5446-5450.

[5] E. Ghadimi, F. Davide Calabrese, G. Peters and P. Soldati, "A reinforcement learning approach to power control and rate adaptation in cellular networks," 2017 IEEE International Conference on Communications (ICC), Paris, 2017, pp. 1-7.

[6] N. Zhao, Y. Liang, D. Niyato, Y. Pei and Y. Jiang, "Deep Reinforcement Learning for User Association and Resource Allocation in Heterogeneous Networks," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6.

[7] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for interference control in OFDMA-based femtocell networks," in Proc. IEEE Veh. Technol. Conf., May 2010, pp. 1–5.

[8] B. Wen, Z. Gao, L. Huang, Y. Tang, and H. Cai, "A Q-learning-based downlink resource scheduling method for capacity optimization in LTE femtocells," in Proc. IEEE. Int. Comp. Sci. and Edu., Aug 2014, pp. 625–628.

[9] H. Saad, A. Mohamed, and T. ElBatt, "Distributed cooperative Q-learning for power allocation in cognitive femtocell networks," in Proc. IEEE Veh. Technol. Conf., Sept 2012, pp. 1–5.

[10] 5G; NR; Physical Channels and Modulation, ETSI TS 138 211 v15.2.0. 3GPP, 2018.

[11] 3GPPP 38.104, Table 5.3.3-1: Minimum guardband [kHz] (FR1) and Table: 5.3.3-2: Minimum guardband [kHz] (FR2)

[12] R. S. Sutton and A. G. Barto, Introduction to Reinforcement Learning, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller. Playing Atari with deep reinforcement learning

[14] 3GPP TR 38.901 Study on channel model for frequencies from 0.5 to 100 GHz.

[15] 3GPP TR 25.942 Radio Frequency (RF) system scenarios