Revista de Informática Teórica e Aplicada - RITA - ISSN 2175-2745
Vol. 31, Num. 2 (2024) 11-19

RESEARCH ARTICLE

Power Allocation in Orthogonal Frequency Division Multiplexing-based Wireless Networks Using Reinforcement Learning

Alocação de Energia em Redes Sem Fio Baseadas em Multiplexação por Divisão de Frequência Ortogonal Usando Aprendizado por Reforço

Hudson Henrique de Souza Lopes[1]*, Flávio Henrique Teles Vieira[1]

Abstract: Energy efficiency is one of the most essential requirements for wireless sensor networks and the internet of things, since batteries are usually the main source of power for these networks. Appropriate techniques based on artificial intelligence can reduce power consumption and provide the required quality of service parameters for the network. In this paper, we approach the challenging problem of allocation of signal transmission power based on the orthogonal frequency division multiplexing technique. We propose to use reinforcement learning-based algorithms to find the optimal policy for allocating power to wireless network devices using a reward function. More specifically, we propose to use the double deep Q-network (DDQN) agent due to its higher learning capacity compared to the Q-learning, deep Q-network (DQN), and the distributed algorithm (DA), a classic algorithm in the literature for power allocation. Simulation results show that the DDQN agent presents promising solutions for power allocation in wireless networks.
Keywords: Reinforcement Learning — Resource Allocation — Wireless Network — Deep Neural Networks

Resumo: A eficiência energética é um dos requisitos mais essenciais para redes de sensores sem fios e a internet das coisas, uma vez que as baterias são normalmente a principal fonte de energia para estas redes. Técnicas apropriadas baseadas em inteligência artificial podem reduzir o consumo de potência e fornecer os parâmetros de qualidade de serviço necessários para a rede. Neste artigo, abordamos o desafiante problema de alocação de potência de transmissão do sinal baseado na técnica de multiplexagem por divisão frequência ortogonal. Propomos a utilização de algoritmos baseados em aprendizagem por reforço para encontrar a política ótima na alocação de potência para dispositivos da rede sem fio utilizando uma função de recompensa. Mais especificamente, propomos a utilização do agente rede Q profunda dupla (DDQN - double deep Q-network) devido à sua maior capacidade de aprendizagem em comparação com a aprendizagem Q (Q-learning), rede Q profunda (DQN - deep Q-network) e o algoritmo distribuído (AD), um algoritmo clássico da literatura para alocação de potência. Os resultados das simulações mostram que o agente DDQN apresenta soluções promissoras para a alocação de potência em redes sem fios.
Palavras-Chave: Aprendizagem por Reforço — Alocação de Recurso — Rede Sem Fio - Redes Neurais Profundas

[1] *School of Electrical, Mechanical and Computer Engineering (EMC), Federal University of Goiás (UFG), Goiânia, Goiás, Brazil*
***Corresponding author**: hudson_lopes@ufg.br
DOI: https://doi.org/10.22456/2175-2745.130630 • **Received:** 06/03/2023 • **Accepted:** 27/06/2024

1. Introduction

In wireless mobile communication systems, radio resource allocation is crucial to improve data transmission efficiency. The base station (BS) can optimize the transmission power allocation by selecting user equipment (UE) in each transmission time interval (TTI) according to the channel state information (CSI), in order to maximize the throughput total system transmission. The data exchange volume in wireless sensor networks (WSN) is growing in recent years due to the expansion of internet of things (IoT) applications. Resource availability and heterogeneous IoT communication lead to energy consumption constraints, as devices in IoT networks are usually powered by batteries, therefore, an important element in device management is energy efficiency (1).

The allocation of resources in wireless networks is typically described as a series of optimization problems, which aim to enhance certain parameters (throughput or packet delay) constrained by limited resources (eg., transmission time, bandwidth, or power). The mathematical modeling of resource allocation in short time intervals assumes that the communication channel is quasi-static. These problems are generally defined as deterministic optimization problems (2). We formulate the problem of allocating power from the BS to the UE considering state transitions, where the packets arrival and the CSI are random variables. These problems are generally referred to as stochastic optimization problems. In this article, we use reinforcement learning (RL) and deep reinforcement learning (DRL) agents, branches of artificial intelligence, that use deep neural networks (DNNs) to mitigate power allocation problems.

DRL is a rapidly growing field of artificial intelligence that aims to solve decision-making problems in complex environments. It combines reinforcement learning techniques with deep neural networks, allowing autonomous agents to learn to make optimal decisions based on past experiences. DRL is based on the idea that an agent can learn to maximize a reward through a sequence of actions. The agent takes an action, receives a reward, and uses that information to improve its future strategy. This is repeated over time, resulting in an optimal decision-making policy. DNNs are used to model the agent's policy and value function, which estimate the expected reward for a given action in a given state. During training, the neural networks are updated using gradients based on the observed rewards.

The main contributions of this work are summarized below:

- RL and DRL algorithms were applied to the problem of power allocation from the BS to the UEs without any prior knowledge of the CSI and the packets arrival.
- Modeling the problem using state transitions and a reward function without requiring a mathematical expression in a closed form for the power allocation problem, where the agents learning process occurs via interaction with the stochastic environment.
- The RL and DRL algorithms and the distributed algorithm (DA) from the literature validate their performance through extensive network simulations. The results show that DRL significantly outperforms Q-learning and the DA, which are not based on DNN.

The remaining work is organized as follows: Section 2 reviews the main works related to this article; Section 3 presents the wireless communication system scenario considered; Section 4 describes the selected power allocation model; Section 5 formulates the power allocation problem through state transitions; Section 6 characterizes the RL and describes the motivation to migrate to DRL; Section 7 discusses the results obtained by agents that learn through reward functions and the DA literature algorithm; and finally, Section 8 summarizes the conclusions reached.

2. Related Works

In the last decade, promising results have been achieved by research when attempting to tackle the reported challenges for resource allocation in modern wireless networks, such as 5th (5G) and 6th (6G) generation networks. In (1), a method designed for a cluster-based communication protocol to improve the energy efficiency in Wireless Sensor Networks (WSN) by considering the IoT objects arrangement is proposed. The results reveal that the proposed method optimizes successfully the constraints associated with IoT scalability and WSN's energy efficiency i.e. the energy and time. Mauricio, F. et al. (3) study the radio resource allocation problem involving energy efficiency optimization with specific requirements of quality of service (QoS) in multi-service scenarios. The work goes further by using RL to perform power allocation. KOO, J. et al. (4) address the problem of resource allocation in a network slicing scenario using an approach based on Markov decision process (MDP) and DRL. The system model uses real and synthetic 4G traffic data and aims to allocate bandwidth and virtual machine resources. However, the authors disregard the allocation of power to the UEs, a factor that affects the communication channel's quality.

In (5), the authors present a DRL approach for creating an intelligent IoT network by blending DRL and software-defined networks. The solution allows training DRL agents to select optimal allocation decision policies for task assignments and scheduling in real time. The performance evaluation shows lower-latency communication and increased energy efficiency. However, the authors do not address the allocation of radio resources in the IoT network. LIU, Y. et al. (6) formulate the resource allocation problem as a restrictive MDP and solve it by using restrictive RL, which assumes that user traffic and mobility patterns are unknown to the algorithms; the algorithms extrapolates and learns from the network without this prior knowledge. This is one of the reasons why learning-based approaches that incorporate exploring the environment for good solutions outperform traditional approaches, which ignore the randomness of the system and are based only on the closed mathematical formulation of the problem.

In (7), the authors proposed a DRL-based NS technique. The DQN algorithm was more specifically used to find the resource allocation policy maximizing the throughput while satisfying the QoS requirements in B5G systems. The authors employed a technique for eliminating undesirable actions that cannot satisfy the QoS requirements. Accordingly, the DRL agent exploration was directed toward desirable actions, improving the chances of making the optimal decision in resource allocation and enhancing the convergence speed in DRL training. In (8), the authors present an algorithm, namely Network Slicing based on Reinforcement Learning (NSRL), that combines the proposed reinforcement learning-based resource allocation with an approach based on reservation and sharing of resources among the slices where each RL agent acts in one slice.

DRL has been successfully applied in various domains, including games, robotics, and simulations. However, deep reinforcement learning still faces some challenges, including instability during training and the need for large amounts of data to produce optimal policies. In addition, the interpretability of policies learned by deep neural networks is still an active research topic. In summary, DRL is an evolving and promising area of research that has already produced impressive results in various domains. However, in this work, we try to overcome some challenges and extend its applicability to resource allocation in wireless networks by improving some QoS parameters. Our main goal is to flexibly and efficiently solve the power allocation issue to improve the fulfillment of the desired QoS requirements with the least amount of power. To achieve this, we propose using DRL algorithms that adapt and learn from the environment without any prior knowledge of precise mathematical models.

3. System Model

This work adopts the downlink direction as its wireless communication system model, which consists of a small cell with a BS at its center as shown in Figure 1. Resource allocation is performed at each TTI with the UE positions in the coverage area changing over time, resulting in variable CSIs across the distribution Rayleigh channel fading. The packet's arrivals are modeled using the Poisson distribution.

Figure 1. Wireless communication system model

This paper considers orthogonal frequency division multiplexing (OFDM) as a technique in LTE downlink transmission, as it allows the simultaneous transmission of different data packets, assigning different subcarriers to the user. In the time domain, the downlink frame duration is 10 ms. These frames are divided into 10 subframes, where each subframe represents a TTI of 1 ms.

4. Power Allocation

The water-filling method aims to allocate the BS signal power to the UEs in such a way as to maximize the sum of the network data rate and meet the constraint of allocating power with positive values to the UEs, keeping within the BS power availability (2). The water-filling method was chosen because it solves the optimization problem described by Equations (1), (2), and (3) with linear computational complexity by solving a system with $(Z + 1)$ equations and $(Z + 1)$ unknowns. This computational complexity is lower than that of other methods in the literature often used to solve the problem, such as genetic algorithms.

$$\max_{p} \quad \sum_{i=1}^{Z} log\left(1 + p_i.\frac{||h_i(t)||^2}{\sigma_z^2}\right), \tag{1}$$

$$\text{Subject to} \quad \sum_{i=1}^{Z} p_i \leq P_{bs}, \tag{2}$$

$$p_i \geq 0, \quad i = 1, 2, ..., Z. \quad P_{bs} \geq 0, \tag{3}$$

where Z is the number of UEs; P_{bs} is the BS power and $p = p_1, p_2, ..., p_Z$ is the power allocated to the UEs. The expression $\frac{||h_i(t)||^2}{\sigma_z^2}$ represents the CSI and is given by the Rayleigh fading distribution. By applying Lagrange's Dual method ($\mathscr{L}$) the objective function (1) can be reduced to:

$$\mathscr{L}(p,\beta) = \sum_{i=1}^{Z} log\left(1 + p_i.\frac{||h_i(t)||^2}{\sigma_z^2}\right) - \beta(\sum_{i=1}^{Z} p_i - P_{bs}), \tag{4}$$

where β is the Lagrange multiplier. A solution to Equation (4) dual problem is obtained by setting the gradient to zero, i.e., $\nabla\mathscr{L}(p,\beta) = 0$, as follows:

$$\frac{\partial \mathscr{L}(p,\beta)}{\partial p} = \frac{\frac{||h_i(t)||^2}{\sigma_z^2}}{1 + p_i.\frac{||h_i(t)||^2}{\sigma_z^2}} - \beta = 0, \tag{5}$$

$$p_i = \frac{1}{\beta} - \frac{1}{\frac{||h_i(t)||^2}{\sigma_z^2}}. \tag{6}$$

The β value choice determines the "water level" for the constraint of the sum of power allocated to the UEs, i.e., the total power consumed by the BS. Therefore, the power allocation for all UEs can be rewritten as:

$$p_i(t) = \max\left\{0, \frac{1}{\beta_t} - \frac{\sigma_z^2}{||h_i(t)||^2}\right\}, \forall i = 1, 2, ..., Z. \tag{7}$$

As the amount of power allocated to UEs with better channel quality is larger, the algorithm deems it unnecessary for some UEs to be powered, given the considerably high noise level. The water-filling algorithm regulates the BS signal power allocation. The idea in modern wireless networks is that the availability of the communication of UEs should be high. However, UEs with a worse channel quality consume a larger amount of signal power to have a minimum data throughput. The algorithm blocks some UEs by limiting communication only to UEs with better channel quality so that the data rate is as high as possible. This mechanism is especially crucial for providing some QoS requirements in wireless networks.

5. Problem Formulation

In this work, we propose the application of RL and DRL algorithms to solve the problem of power allocation in OFDM-based wireless networks. In this RL approach, system states are considered to be represented by the number of packets that are waiting for transmission in buffer $q(t)$, and the action is given by the variable β that is related to the power allocated to the UEs. The reward function consists of minimizing the $\frac{q(t)}{\lambda}$ delay and the sum of the power that is allocated to the UEs ($\sum_{i=1}^{Z} p_i$) where λ is the parameter of the Poisson distribution for the package's arrival.

The major elements for the agents application to the power allocation problem are described below:

System state: the number of packets awaiting transmission in the t-th subframe is considered as the system state $s(t)$, that is, the packets present in the buffer:

$$s(t) = \{q(t)\}. \tag{8}$$

An R-sized buffer stores a maximum of R packets for all UEs. The system states are obtained while taking into account (R + 1) possible states, including zero. Hence, the buffer state changes with the arrival or departure of packets at each subframe.

Control policy: given the random distribution of the CSI of the UEs, control action is the power allocation using Eq.(7) through the β value, which is discretized in two decimal places ∈ (0.01, 1):

$$a(t) = \{\beta\}. \tag{9}$$

State transition: given the state of the system, the CSI of the UEs, and the control action of the t-th subframe, the state transition is represented by the dynamic queue equation given by:

$$q(t+1) = max\{0, q(t) - d(t)\} + c(t), \tag{10}$$

where: $q(t+1)$ is the new state of the system $s(t+1)$ and $c(t)$ is the number of packets arriving in the t-th subframe. It is assumed that $c(t)$ follows the Poisson distribution with parameter λ. Thus, there is an average of λ packets in each subframe. The channel capacity $d(t)$ represents the amount of the packet transmitted in the t-th subframe given by:

$$d(t) = \left(\frac{\sum_{i=1}^{Z} L_i \times log_2(1 + \frac{P_i.||h_i||^2}{\sigma_z^2})}{B} \right), \tag{11}$$

where L_i is the bandwidth; and B is a packet's amount of bits. We considered 1024 bytes packet sizes, the latter being based on the IEC-61850 standard for medium and high voltage power distribution (9). This paper assumes an equal bandwidth allocation for all UEs.

Reward function: According to Little's Law, the average packet delay is given by (10):

$$W = \frac{\bar{Q}}{\lambda} = \lim_{T\to\infty} E\left(\frac{1}{T}\sum_{i=1}^{T}\frac{q(t)}{\lambda}\right), \tag{12}$$

where $\bar{Q}$ is the average number of packets awaiting transmission. Therefore, the reward function is given by:

$$r(t) = -\sum_{i=1}^{Z} p_i - \alpha \times W, \tag{13}$$

where α is the weight of the average packet transmission delay. In general, RL agents learn with the main objective of maximizing the reward function. In this work, the main objective is to reduce the average delay by consuming as little power as possible; the reward function is negative in the equation, thus minimizing its value. The weight α is used to give priority to reducing the average delay.

6. Reinforcement Learning

Value-based algorithms are used to determine the agent's value function. This value function is then used to implicitly and greedily obtain an optimal policy. There are two types of value-based functions: the value functions $V^{\pi}(s)$ and the state-action function $Q(s(t), a(t))$. Both represent the expected cumulative discontinuous reward received when taking action $a(t)$ in state $s(t)$. These functions are quite important, as they represent the link between the mathematical formulation of the MDP and the RL. The value function $V^*(s)$ and the state-action function $Q^*(s(t), a(t))$ are obtained by Bellman's optimality equation (11):

$$V^*(s) = \max_{a(t)}[r_t(s(t), a(t)) + \gamma E_\pi V^*(s(t+1))]. \tag{14}$$

$$Q^*(s(t), a(t)) = r_t(s(t), a(t)) + \gamma E_\pi \times [\max_{a(t+1)} Q^*(s(t+1), a(t+1))], \tag{15}$$

where γ is the discount factor ∈ (0,1); and $r_t(s(t), a(t))$ is the reward for taking action $a(t)$ in state $s(t)$. The main objective of the MDP is to obtain the optimal policy π^*, i.e., mapping states to optimize actions. For the value function, the policy is given by:

$$\pi^* = \arg\max_{a(t)} = E[\sum_{t=1}^{T} \gamma r_t(s(t), \pi(s(t)))]. \tag{16}$$

For the Q function, the optimal policy becomes:

$$\pi^*(s) = \arg\max_{a} Q^{\pi*}(s(t), a(t)). \tag{17}$$

In RL, Q-learning is the most used algorithm to address MDPs. It obtains optimal function $Q(s(t), a(t))$ values by iteratively using the Bellman equation update rule (11):

$$Q(s(t), a(t)) = Q(s(t), a(t)) + \omega[r_t(s(t), a(t)) + \gamma \max_{a(t+1)} Q^*(s(t+1), a(t+1)) - Q(s(t), a(t))], \tag{18}$$

where ω is the learning rate.

6.1 Double Deep Q-Network (DDQN)

The Q-learning algorithm is based on building a table for the values of the Q function. Due to this reason, when the state space and the action space become large as in the cases commonly encountered in the resource management problems of modern wireless systems obtaining the optimal policy could be extremely time-consuming.

To solve this problem, Deep Q-Network (DQN), which inherits the advantages of Q-learning and Deep Learning (DL) techniques are commonly used. The main idea is to replace the table of the Q-learning algorithm with a DNN that approximates the Q-values.

The DNN is also called a universal approximation function and is denoted by $Q(s(t), a(t)|\Theta)$, where Θ represents the parameters or weights of the DNN. To increase the stability of the DQN, another neural network is used, called the target Q network, whose weights $\Theta^{'}$ will be periodically updated to follow those of the main Q neural network (11).

The DQN algorithm is iteratively optimized by updating the Θ weights of its DNN to minimize the following Bellman loss function:

$$L(\Theta_t) = E_{s(t),a(t),r_t,s(t+1)}[r_t(s(t), a(t)) + \gamma \max_{a(t+1)} Q(s(t+1), a(t+1)|\Theta^{'}) - Q(s(t), a(t)|\Theta)]^2, \tag{19}$$

where $\Theta^{'}$ are the weights of the target Q network.

The DQN algorithm tends to overestimate Q-values, which can degrade the training process and lead to sub-optimal policies. The overestimation results from the positive bias caused by the maximal operation employed in the Bellman equation. Specifically, the root cause is that the same training transitions are used in the selection and evaluation of an action (11). As a solution to this problem, we propose to use the Double DQN (DDQN) technique, where two functions are employed for the Q-value, one to select the best action and the other to evaluate the best action. The action selection is still based on the Θ weights, while the second $\Theta^{'}$ weights are used to evaluate the value of this policy as shown in Figure 2. Therefore, as in conventional Q-learning, the value of the policy is still estimated based on the current Q-value. The parameters of weights $\Theta^{'}$ are updated by Θ (11).

The DDQN Algorithm uses the following modified Bellman loss function to update its weights:

$$L(\Theta_t) = E_{s(t),a(t),r_t,s(t+1)}[r_t(s(t), a(t)) + \gamma Q(s(t+1), \arg\max_{a(t+1)} Q(s(t+1), a(t+1)|\Theta), \Theta^{'}) - Q(s(t), a(t)|\Theta)]^2. \tag{20}$$

The DDQN architecture approaches three key techniques to improve convergence in learning (12):

- Experience replay: Experience samples generated by the agents are stored in an experience replay buffer, and then a mini-batch is randomly sampled from the buffer to train the neural network. This strategy breaks the correlation between the training samples.
- Target Q-network: As shown in Figure 2, the main network is trained using the loss function from Equation (20), while a target network is run to generate experience samples for training purposes. The main Q-network and the target Q-network have the same architecture but with two different sets of weights: Θ and $\Theta^{'}$, respectively. The weights of the target network are updated periodically every τ step with the same weights as the main network.
- Decoupling in action selection and evaluation: The target Q-network is used to generate the Q-values that will be used to calculate the loss during training, while the main Q-network is used to select which is the best action to take to the next state. By decoupling action selection from evaluation, the risk of overestimating Q-values can be greatly reduced. In other words, in case the main Q-network overestimates the action, the target Q-network would generate an appropriate value.

Algorithm 1 illustrates the training process in more detail.

7. Results and Discussion

In this work, simulations were performed using the following software and hardware configurations: Matlab software, 1.00 GHz Intel Core i5-1035G1 processor; 8 GB of RAM without a dedicated graphic card. The neural network of the two methods (DQN and DDQN) was implemented with three

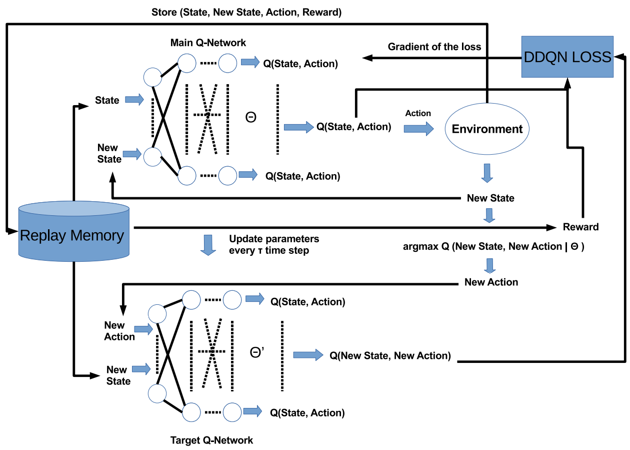


Figure 2. Double DQN architecture

Algorithm 1: DDQN Training Process

Initialize the weights of the main Q-network Θ = random;
Initialize the target Q-network weights $\Theta^{'} = \Theta$;
for *t from 1 to t_{max}* **do**
  if *$mod(t,\tau) == 0$* **then**
    Updates the weights of the target network $\Theta^{'} = \Theta$;
  end
  Exploration condition based on greedy policy ε;
  if *rand(0,1) $< \varepsilon$* **then**
    Sample a random action a(t);
  else
    a(t) = $\arg\max_{a(t)} Q(s(t), a(t)|\Theta)$;
  end
  Apply a(t) to the environment and observe r(t) and s(t+1);
  Add recent experience <s(t),a(t),r(t),s(t+1)> in the replay buffer;
  Sample a mini-batch from the replay buffer;
  Updates Θ according to the loss function Equation (20);
end

fully connected hidden layers with 500 neurons in each and the leaky ReLU activation function. After several exhaustive simulations, the hyperparameters were set with 0.5 learning rate, 0.1 random action choice chance, 0.9 discount factor, 256 mini-batch size, and 10000 replay buffer size. The complete set of simulation parameters is provided in Table 1.

Figures 3-6 shows the RL and DRL algorithms performance (Q-learning, DQN, and DDQN) in BS power allocation as the number of active UEs in the network increases. In terms of reward function values, the convergence performance of the algorithms was analyzed for 10000 iterations of a 1000 TTI simulation. For a larger number of iterations, no significant changes were observed in the results obtained with the agents considered. Figure 3 shows the agents' learning process in managing the allocation of signal transmission power to reduce the average delay in packet transmission, two essential parameters for using wireless networks. The values of the rewards were plotted as a function of the number of increments of the UEs in the training phase. Figure 3 shows that the long-term discounted reward tends to gradually decrease as the number of UEs in training increases. In Figure 3, the higher reward values obtained by the DDQN algorithm compared to the other algorithms indicate its greater learning capacity for the problem because it employs two evaluation functions in its learning method.

A comparison is also performed with the algorithm of the literature such as the Distributed Algorithm (DA) proposed in (13), an algorithm based on the proportional fairness technique that has as its main objective to allocate more resources to the UEs with worse channel quality. Figure 4 shows the sum of the average power of the transmission signal allocated to the UEs used to increase the transmission channel capacity and send the packets queuing up. Figure 4 also shows that the highest power consumption of the DA algorithm exceeds the total available power of the BS because it does not learn from the environment to allocate power in order to save resources. The DDQN algorithm has the greatest power savings compared to the other algorithms.

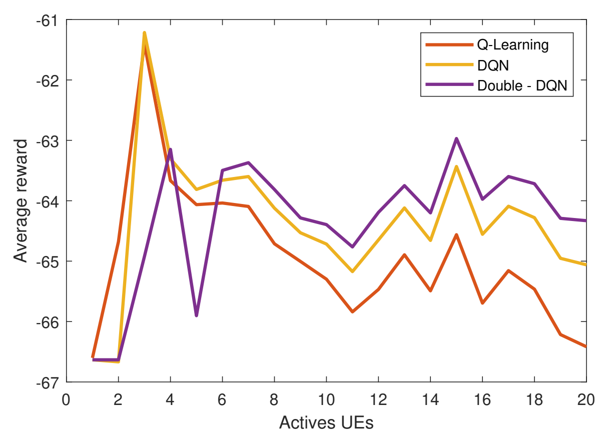


Figure 3. Average reward comparison of different agents.

In this work, a high weight was given to the average delay in the reward function of the RL agents, increasing the priority on reducing the average delay in transmitting packets with the smallest amount of allocated power possible. The results in Figure 5 reveal that the highest power consumption of the DA algorithm shown in Figure 4 did not yield benefits in terms of average delay and packet throughput. Increased numbers of active UEs in the system lead to a surged sum of average allocated power as shown in Figure 4, which causes the average throughput as shown in Figure 5(b) to also rise and remain approximately constant after 7 UEs in the system. Therefore, from 7 UEs and afterward, the algorithms empty

Table 1. Simulation parameters

Parameters	Values
BS power (P_{bs})	10 W
Maximum buffer size (R)	4 packets
TTI	1 ms
Number of OFDM symbols per TTI	7
Bandwidth	20 MHz
Channel modeling	Rayleigh distribution
Chance of random choice (ε)	0.1
Learning rate (ω)	0.5
Discount factor (γ)	0.90
Weight on average delay (α)	50
Package arrival rate (λ)	Possoin distribution
Hidden layers of dense neural network	3
Number of neurons in each layer	500
Activation function of each hidden layer	Leaky ReLU
Number of bits in a packet (B))	8192
Training iterations	10000
Simulation time	1000 TTI

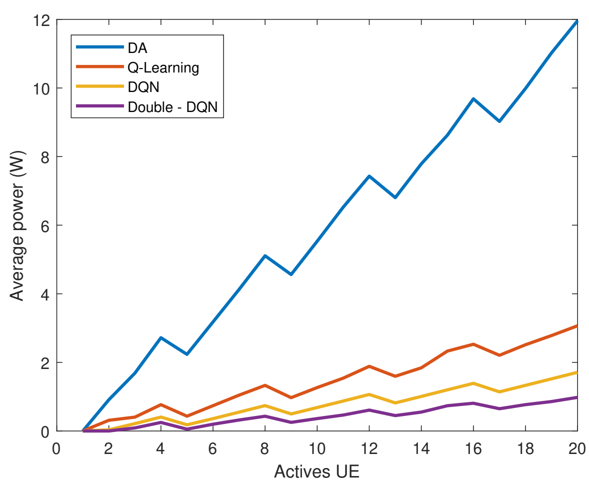
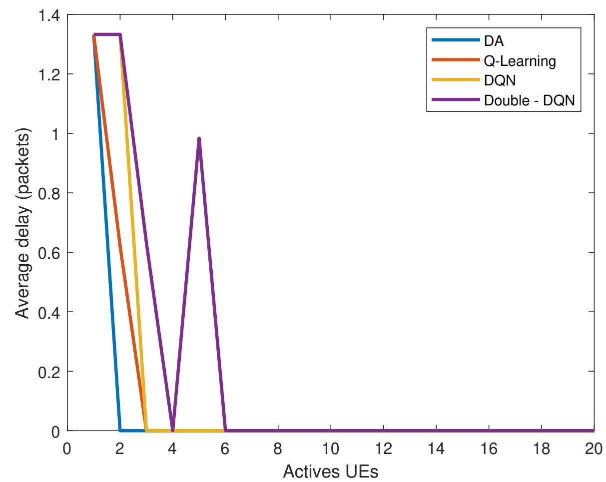


Figure 4. Sum of the average power allocated to the UEs.

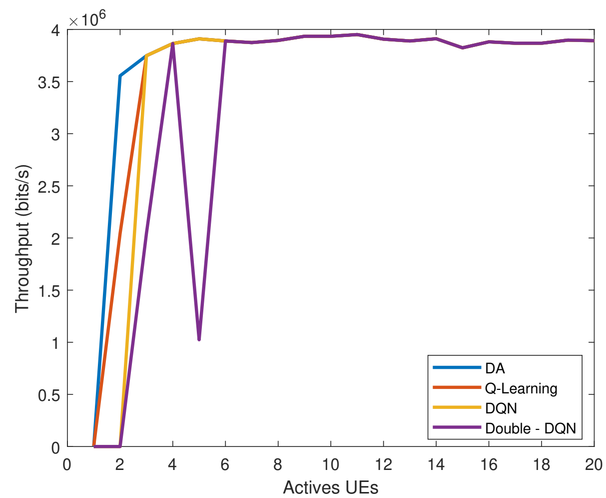
the buffer, reducing the average delay as shown in Figure 5(a). On average, considering all power allocation algorithms used in the simulation, the DDQN algorithm noticeably obtained the same results in packet transmission delay and throughput from 7 active UEs in the network despite having the lowest power consumption.

Energy efficiency (EE) has emerged as a key performance indicator for future networks with the shift toward optimized communications. It can be improved by using various strategies, including infrastructure planning and development, network structure, energy capture, and radio resource allocation (14). This work considers the EE metric as the ratio between the average transmission rate and the total average transmission power consumed. As shown in Figure 6(a), the DDQN agent presents the best EE results. Another indicator analyzed is the loss of packets in the buffer. Figure 6(b) shows that increased numbers of active UEs in the system, push the packet's arrival beyond buffer capacity, leading to packet loss. Power allocation algorithms reveal packet losses that grow in tandem with the number of active UEs in the network. However, from 7 UEs in the system, packet losses become practically equal for all the power allocation algorithms considered.

Figure 5. Average delay and average throughput in packet transmission.



(a) Average delay.

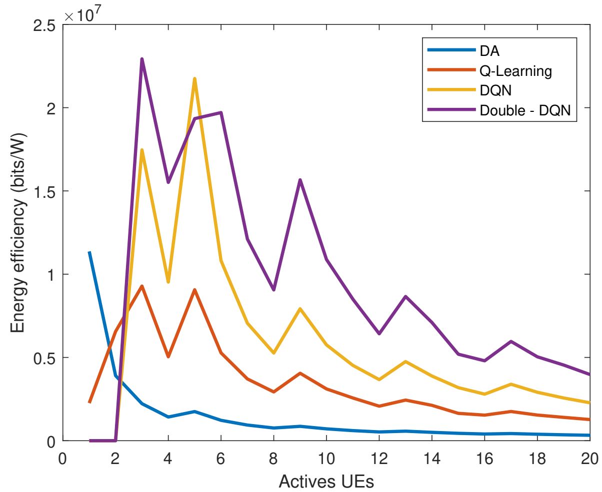


(b) Average throughput.

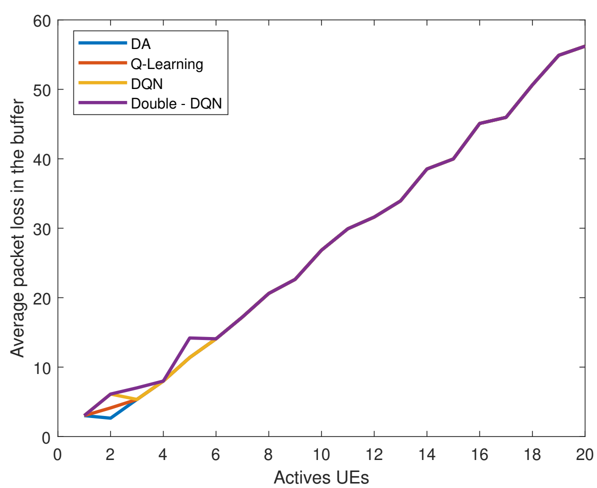
8. Conclusion

This article proposes the application of RL and DRL agents to solve the problem of power allocation of the transmission signal from the BS to the UEs. The problem was modeled

Figure 6. Packet transmission energy efficiency and average packet loss in the buffer.



(a) Average energy efficiency.



(b) Average packet loss.

through state transition regardless of prior knowledge of the transitions probabilities between states; that is, the agent adaptively learns based on the observations of the system's random states. Extensive simulations showed that the DDQN agent outperforms the others in terms of EE given its enhanced ability to learn from the environment via the proposed reward function. In a scenario with more than 7 active UEs in the network, the DDQN agent's efficient power consumption afforded greater savings.

The performances of the DRL algorithms were compared with the agent Q-learning and the DA algorithm present in the literature (13). The DA algorithm does not have the ability to learn from the environment and does not allocate power efficiently. The DRL methods are considered promising techniques for resource allocation in wireless networks, as pointed out in the literature (4), (6) and (15). Due to its great learning potential, interesting results are obtained by monitoring the environment, without a priori knowledge of the system's statistics. The simulation results presented in this article confirm the DDQN algorithm's learning efficiency for allocating power to the UEs. Finally, in future works, we intend to adapt the scenario in order to consider next-generation wireless mobile networks.

Acknowledgements

The authors would like to thank Fundação de Amparo à Pesquisa no Estado de Goiás (FAPEG), the Centro de Excelência em Inteligência Artificial (CEIA), the Centro de Excelência em Redes Inteligentes Sem Fio e Serviços Avançados (CERISE) and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) for their support and funding in the development of the research.

Author contributions

All authors have contributed equally to the development of this work.

References

[1] THANGARASU, G. et al. An efficient energy consumption technique in integrated wsn-iot environment operations. In: *2019 IEEE Student Conference on Research and Development (SCOReD)*. [S.l.: s.n.], 2019. p. 45–48.

[2] DING, Z. (Ed.). *Applications of Machine Learning in Wireless Communications*. [S.l.]: Institution of Engineering and Technology, 2019. 371-405 p. (Telecommunications).

[3] Mauricio, F. et al. Resource allocation for energy efficiency and qos provisioning. *Journal of Communication and Information Systems*, v. 34, n. 1, p. 224–238, Oct. 2019. Disponível em: ⟨https://jcis.sbrt.org.br/jcis/article/view/665⟩.

[4] KOO, J. et al. Deep reinforcement learning for network slicing with heterogeneous resource requirements and time varying traffic dynamics. In: *2019 15th International Conference on Network and Service Management (CNSM)*. [S.l.: s.n.], 2019. p. 1–5.

[5] SELLAMI, B. et al. Deep reinforcement learning for energy-efficient task scheduling in sdn-based iot network. In: *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*. [S.l.: s.n.], 2020. p. 1–4.

[6] LIU, Y.; DING, J.; LIU, X. Resource allocation method for network slicing using constrained reinforcement learning. In: *2021 IFIP Networking Conference (IFIP Networking)*. [S.l.: s.n.], 2021. p. 1–3.

[7] SUH, K. et al. Deep reinforcement learning-based network slicing for beyond 5g. *IEEE Access*, v. 10, p. 7384–7395, 2022.

[8] LOPES, H. S.; ROCHA, F.; VIEIRA, F. Deep reinforcement learning based resource allocation approach for wireless networks considering network slicing paradigm. *Journal of Communication and Information Systems*, v. 38, n. 1, p. 21–33, Feb. 2023. Disponível em: ⟨https://jcis.sbrt.org.br/jcis/article/view/836⟩.

[9] WONG, T. J.; DAS, N. Modelling and analysis of iec 61850 for end-to-end delay characteristics with various packet sizes in modern power substation systems. In: *5th Brunei International Conference on Engineering and Technology (BICET 2014)*. [S.l.: s.n.], 2014. p. 1–6.

[10] KLEINROCK, L. *Theory, Volume 1, Queueing Systems*. USA: Wiley-Interscience, 1975. v. 1. ISBN 0471491101.

[11] ALWARAFY, A. et al. Deep reinforcement learning for radio resource allocation and management in next generation heterogeneous wireless networks: A survey. *CoRR*, abs/2106.00574, 2021. Disponível em: ⟨https://arxiv.org/abs/2106.00574⟩.

[12] GU, B. et al. Deep multiagent reinforcement-learning-based resource allocation for internet of controllable things. *IEEE Internet of Things Journal*, v. 8, n. 5, p. 3066–3074, 2021.

[13] Abdel-Hadi, A.; Khawar, A.; Clancy, T. C. Optimal downlink power allocation in cellular networks. *Physical Communication*, v. 17, p. 1–14, 2015.

[14] BUZZI, S. et al. A survey of energy-efficient techniques for 5g networks and challenges ahead. *IEEE Journal on Selected Areas in Communications*, v. 34, n. 4, p. 697–709, 2016.

[15] ZHOU, H.; ELSAYED, M. H. M.; EROL-KANTARCI, M. RAN resource slicing in 5g using multi-agent correlated q-learning. In: *32nd IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2021, Helsinki, Finland, September 13-16, 2021*. [S.l.]: IEEE, 2021. p. 1179–1184.