CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Deep reinforcement learning for combined coverage and resource allocation in UAV-aided RAN-slicing
Authors	Bellone, Lorenzo;Galkin, Boris;Traversi, Emiliano;Natalizio, Enrico
Publication date	2023-09-27
Original Citation	Bellone, L., Galkin, B., Traversi, E. and Natalizio, E. (2023) 'Deep reinforcement learning for combined coverage and resource allocation in UAV-aided RAN-slicing', 2023 19th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Pafos, Cyprus, 19-21 June, pp. 669-675. doi: 10.1109/DCOSS-IoT58021.2023.00106
Type of publication	Conference item
Link to publisher's version	10.1109/dcoss-iot58021.2023.00106
Rights	© 2023, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-15 01:50:40
Item downloaded from	https://hdl.handle.net/10468/15182

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Deep Reinforcement Learning for Combined Coverage and Resource Allocation in UAV-Aided RAN-Slicing

Lorenzo Bellone*, Boris Galkin†, Emiliano Traversi*‡, and Enrico Natalizio*§
* Autonomous Robotics Research Center - Technology Innovation Institute, UAE
† Wireless Communications Group - Tyndall National Institute, Ireland
‡ LIPN, UMR CNRS 7030 - Université Sorbonne Paris Nord, France
§ LORIA, UMR CNRS 7503 - Université de Lorraine, Nancy, France
lorenzo.bellone@tii.ae, boris.galkin@tyndall.ie, emiliano.travers@lipn.fr, enrico.natalizio@tii.ae

***Abstract*—Network slicing is a well assessed approach enabling virtualization of the mobile core and radio access network (RAN) in the emerging 5th Generation New Radio. Slicing is of paramount importance when dealing with the emerging and diverse vertical applications entailing heterogeneous sets of requirements. 5G is also envisioning Unmanned Aerial Vehicles (UAVs) to be a key element in the cellular network standard, aiming at their use as aerial base stations and exploiting their flexible and quick deployment to enhance the wireless network performance. This work presents a UAV-assisted 5G network, where the aerial base stations (UAV-BS) are empowered with network slicing capabilities aiming at optimizing the Service Level Agreement (SLA) satisfaction ratio of a set of users. The users belong to three heterogeneous categories of 5G service type, namely, enhanced mobile broadband (eMBB), ultra-reliable low-latency communication (URLLC), and massive machine-type communication (mMTC). A first application of multi-agent and multi-decision deep reinforcement learning for UAV-BS in a network slicing context is introduced, aiming at the optimization of the SLA satisfaction ratio of users through the joint allocation of radio resources to slices and refinement of the UAV-BSs 2-dimensional trajectories. The performance of the presented strategy have been tested and compared to benchmark heuristics, highlighting a higher percentage of satisfied users (at least 10.5% more) in a variety of scenarios.**

I. Introduction

The 5th Generation (5G) New Radio (NR) is expected to support a large variety of vertical use cases (e.g., autonomous vehicles, industrial Internet of Things), which come with a broader range of requirements with respect to the traditional mobile broadband services. The NGNM 5G White Paper [1] groups these vertical use cases into three categories: i) enhanced Mobile Broadband (eMBB), ii) massive Machine Type Communication (mMTC), and iii) Ultra-Reliable Low-latency Communication (URLLC). The compliance with the heterogeneous set of requirements demanded by these services is proven hard to achieve in the monolithic architecture implemented in the 4G system [2]. A partitioning of the physical network into several virtual networks looks to be the most suitable approach to provide a customized service to each application and to limit the operation expenses. This design is well known as *network slicing*.

The concept of network slicing has been investigated for some time. With the purpose of providing multiple core networks over the same network infrastructure, DECOR and eDECOR [3] have been already used in legacy LTE networks. However, the slicing of Radio Access Network (RAN) was not considered in these works yet. Since then, the 5G network slicing concept evolved, by providing a better modularization and flexibility of the network functions. RAN slicing is now of great interest for the literature, given the ability to support resource isolation among different slices, providing a way to allocate radio resources to the connected User Equipments (UEs) based on the slices they belong to. Several RAN slicing approaches have been proposed in the past few years. Traditional strategies ([4], [5]) mainly deal with orthogonal resource allocation in time and/or frequency domain, such that the isolation between different services is guaranteed. Their aim in mainly oriented at allocating virtual Resource Blocks (vRBs) to UEs for intra-slice scheduling and to map the allocated vRBs to Physical Resource Blocks (PRBs) at a second stage. This two level scheduling enables high scalability and dynamicity in the resource management.

Machine Learning (ML) techniques have also been extensively considered, given their capability to find patterns within the huge amount of data that is exchanged in cellular networks. In the literature, two main categories of ML strategies for network slicing have been exploited: i) slice admission control, and ii) cross slice resource allocation. In the first category, the network operator has to decide whether to accept or refuse slice creation requests coming with a resource requirement and a service level agreement demand. This decision has to be made according to the number of resources already allocated and to the monetary value of each slice. In [6], several RL-based slice admission control approaches are examined. Deep Q-Network was proven to outperform other RL techniques to maximize network operator's cumulative revenue. In the cross slice resource allocation category, instead, the dynamicity of the users is taken into account. The resources are not statically allocated to the slices, but they change over time, according to the dynamic requirements of the users. Bega et al. [7]

introduce "DeepCog", a mobile traffic data analytic tool that is tailored at solving a demand forecasting problem. They are able to predict the capacity that each slice will need in future time slots by the design and training of a Deep Neural Network (DNN) that takes as input the current traffic associated to a slice at a given base station. Although this work is mainly oriented at fostering the dynamic allocation of resources to the slices, they do not provide a new algorithm to properly schedule the resources given this prediction. In [8], instead, the authors rely on a end-to-end resource allocation framework. They design a Reinforcement Learning (RL) algorithm based on A2C with an LSTM neural network, which captures the packets exchanged by the slices at each time-slot. This input is then converted to an action, based on the radio allocation for each slice for the next allocation window.

These recent works have proven how ML might enable more powerful strategies for resource allocation in network slicing, exploiting the learning of users' behavioral patterns to predict their requirements and properly manage the radio allocation over time. These reasons motivate our willingness to exploit similar strategies in an Unmanned Aerial Vehicle (UAV)-assisted next generation cellular network.

Many works have already proven how UAVs can foster the capabilities of next generation wireless networks through their easy deployability and strong Line-of-Sight communication links, as meticulously described in [9]. However, there is still a limited number of works dealing with UAVs deployment in a RAN slicing context.

A great example is provided by Yang et al. [10], formulating a solution for proactive UAV network slicing. In their work, a Lyapunov-based optimization framework is developed in order to maximize the users' achievable data rates as well as the UAVs' fair coverage. This optimization technique proved to be effective in a slicing scenario, but it requires a global knowledge of the environment in space and time domain. A second relevant contribution for UAV-based network slicing is given by [11], where they propose a resource allocation algorithm for UAV RAN slicing for eMBB and mMTC users co-existence. While they provide a fair solution for the allocation, the authors do not exploit the mobility of UAVs, which has a great impact on the network performance.

Our work aims at providing an online learning framework, where a set of UAV-BSs will try to maximize the SLA satisfaction ratio of the users deployed in the environment. Unlike the previous mentioned works, this optimization is achieved for the first time (to the best of our knowledge) through a multi-agent and multi-decision approach, trying to exploit the placement of the base stations and the resource allocation for each slice. In more detail, our contribution can be summarized into three fundamental components:

1) we propose a joint placement and resource allocation framework in a UAV-aided RAN slicing system, where eMBB, URLLC, and mMTC users are deployed in the same environment;
2) a multi-decisional and multi-agent reinforcement learning solution is proposed to address these two problems at the same time, exploiting communication between agents;
3) a fair comparison between our approach and benchmark solutions is presented, showing the benefits of a learning approach in the considered environment.

II. Methodology

A. System Model

We consider a downlink UAV-BS system in which users have to be properly allocated with resource blocks by the UAV-BS in order to meet their downlink requirements. The set of UAV-BS is denoted as $\mathcal{U} = \{u_1, u_2 \dots\} \in \mathbb{R}^2$, where each UAV u is free to move in a two-dimensional space, while the height h is fixed. The assumption of a constant height is motivated by safety regulations, which do not allow drones to fly above certain height limits. The radio bandwidth is split into Physical Resource Blocks (PRBs), consisting of 12 consecutive subcarriers 15KHz wide. Each PRB is 1ms long (known as Transmission Time Interval, TTI), thus carrying 14 symbols per subcarrier [12].

A set of users $\mathcal{G} = \{g_1, g_2, g_3 \dots\} \in \mathbb{R}^2$ is spatially distributed following a given distribution. The UAV-BSs will adapt to the placement of the users that might appear in the environment. In our specific scenario, the users are distributed according to multiple bivariate Gaussian distributions.

Each user can belong to one of the three classes of slices, i.e., *eMBB*, *URLLC*, and *mMTC*, each class having different requirements in terms of desired data rate. In our scenario, each user has probability p_{eMBB} to belong to the eMBB, p_{URLLC} to the URLLC, and p_{mMTC} to the mMTC slices. A graphical representation of the proposed environment is shown in Fig. 1.

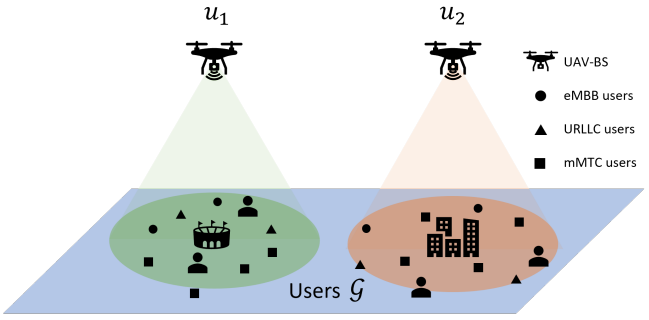


Fig. 1: UAV-BSs hovering on ground users to provide wireless coverage to multiple slices.

The data rate can be theoretically evaluated starting from the Signal to Interference plus Noise Ratio (SINR) between the base station and the associated users, which depends on the specific channel model. For our work, we took inspiration from the Air-to-Ground channel model presented by Galkin et al. [13] (Eq. 1), where the authors exploit a multi-agent Deep Reinforcement Learning (MADRL) approach within a multi-UAV network scenario, aiming at the energy efficiency optimization through the UAVs trajectories adjustment. The

UAVs are supposed to be equipped with down-tilted antennas with a cone-shaped coverage pattern, while the users are provided with an omni-directional antenna. Furthermore, the wireless channel between UAVs and users will always be LoS.

$$\mathrm{SINR}_{i,j}^{t} = \frac{pc\mu\left(y_j,u_i^t\right)\left(\left(\left\|y_j-u_i^t\right\|\right)^2+(h)^2\right)^{-\alpha/2}}{\sum\limits_{k\in\mathcal{U}\setminus i} pc\mu\left(y_j,u_k^t\right)\left(\left(\left\|y_j-u_k^t\right\|\right)^2+(h)^2\right)^{-\alpha/2}+\sigma^2} \tag{1}$$

In Eq. 1 we report the Signal-to-Interference-and-Noise-Ratio (SINR) observed by user i with respect to UAV j, as illustrated in [13]. In this equation, p represents the UAV transmit power, c is the near-field pathloss, α is the pathloss exponent, and σ^2 is the noise power. Furthermore, the function $\mu(y_j, u_i^t)$ returns the value of the antenna gain, which depends on the coordinates of user (y_j) and UAV (u_i), together with the antenna beamwidth.

Once the SINR of a single user has been evaluated, we can i) associate that user with a base station and ii) map the SINR with the modulation and coding scheme (MCS) and code rate, as shown in Table I.

SINR	Modulation Scheme	Code Rate
< 5.2	QPSK	0.5879
> 5.2 and < 6.1	16QAM	0.3691
> 6.1 and < 7.55	16QAM	0.4785
> 7.55 and < 10.85	16QAM	0.6016
> 10.85 and < 11.55	16QAM	0.4551
> 11.55 and < 12.75	64QAM	0.5537
> 12.75 and < 14.55	64QAM	0.6504
> 14.55 and < 18.15	64QAM	0.7539
> 18.15 and < 19.25	64QAM	0.8525
> 19.25	64QAM	0.9257

TABLE I: Mapping between SINR and MCS [14]

The UAV-user association follows a simple strategy: the user is associated with the base station corresponding with the highest value of SINR, as long as this value is greater than a given threshold, which we fixed at 5dB.

Once the user has been associated, the SINR is mapped to the MCS for the transmission between base station and user, and the data rate can be evaluated through Eq. 2, where bps are the bits per symbol evaluated through the modulation scheme, c is the code rate, 168 is the multiplication between the number of sub-carriers in each resource block (RB) and the number of symbols per each sub-carriers (12 and 14 respectively), and r are the associated resource blocks for that user ([14]).

$$dr = 168 \times \frac{bps \times c \times r}{0.001} \tag{2}$$

The bandwidth of the system is 20MHz, thus the scheduler will allocate 100 PRBs every TTI.

In the considered system, each single UAV-BS is in charge of allocating portion of bandwidth to each slice. This allocation results in a different number of PRBs available per users, according to the slice they belong to. Eventually, a round robin scheduler is in charge of associating the PRBs reserved to one slice to the users belonging to that slice.

The goal of our approach is to maximize the number of users with their Service Level Agreement (SLA) satisfied. In the considered scenario, we simulate this metric as being the number of users whose data rate is greater than a given threshold.

The aim is thus twofold: on one hand, we try to allocate portion of the bandwidth to the slices, by taking into account different parameters, such as the requirements of the users and their position. At the same time, the adopted methodology is able to jointly determine the position that the base station will take. Hence, we solve a multi-task decision problem, by addressing simultaneously the resource allocation and placement of each UAV-BS.

B. Optimization Problem Formulation

The objective of this problem is to maximize the overall users' SLA satisfaction, δ. To achieve this goal, we formalize an optimization problem for a scenario where two UAV-BSs are deployed, where the following variables must be properly defined:

- u is a matrix of dimensions $N \times N$, with N being the number of possible positions for a single base station. This matrix enumerates all the possible placement combination of the two UAVs. u_{ij} is a binary variable that is equal to one, if the first UAV is in position i and the second UAV is in position j, with $i, j, \in \{1, \ldots, N\}$;
- bw is a 3×2 matrix. It indicates the bandwidth that each base station allocates for *eMBB*, *URLLC*, and *mMTC* users, respectively. $bw_{em,b}$ is a non negative variable that represents the fraction of bandwidth that the UAV b (with $b \in \{1, 2\}$) allocates to eMBB users. A analogous definitions holds for $bw_{ur,b}$ and $bw_{mm,b}$;
- δ is a vector of dimensions $|\mathcal{G}|$. δ_g is a binary variable that is equal to one if the SLA of user g is satisfied.

Furthermore, for sake of readability, we define an additional parameter G_{conn}(slice,b) (with slice $\in \{em, ur, mm\}$) that represents the number of users associated to base station b than belong to a specific slice. This parameter is needed to perform a round robin scheduling after the radio resources have been allocated to a given slice. Furthermore, we identify with slice(g) the slice user g belongs to.

The optimization problem can be formally written as:

$$max_{u,bw} \sum_{g\in\mathcal{G}} \delta_g \tag{3}$$

$s.t.$

$$\sum_{b\in|\mathcal{U}|} \sum_{i,j\in\mathcal{U}} u_{i,j} \times bw_{\text{slice}(g),b} \times \text{bps}_g \frac{100}{G_{\text{conn}}(\text{slice}(g), b)}$$

$$\geq th_g \times \delta_g \qquad \forall g \in \mathcal{G} \tag{4a}$$

$$bw_{em,b} + bw_{ur,b} + bw_{mm,b} = 1 \qquad \forall b \in |\mathcal{U}| \tag{4b}$$

$$bw_{em,b} \geq 0; bw_{ur,b} \geq 0; bw_{mm,b} \geq 0 \qquad \forall b \in |\mathcal{U}| \tag{4c}$$

$$\sum_{i,j \in \mathcal{U}} u_{i,j} = 1 \tag{4d}$$

$$\delta_g \in \{0,1\} \qquad \forall g \in \mathcal{G} \tag{4e}$$

$$u_{i,j} \in \{0,1\} \qquad \forall i,j \in \{0,1,...,|\mathcal{U}-1|\} \tag{4f}$$

Constraint (4a) defines the data rate that is achievable by one user and compares it to the data rate threshold in order to associate the proper δ_g. The element "bps" is the pre-computed function presented in Eq. 2. The achieved bit per symbols are then multiplied by the number of resource blocks allocated to that user. This number is retrieved by dividing the available resources allocated to the users' slice ($bw_{\text{slice}(g),b} \times 100$) by the number of users belonging to the same slice that are associated to the same base station as the considered user (G_{conn}(slice,b)). Constraint (4b) specifies that, for each base station, the sum of the portions of allocated bandwidth should be equal to one.

C. Multi-Agent and Multi-Decision Learning Approach

In preliminary tests, we observed that the solution of the presented mathematical model requires a high computation time. As an alternative, we propose a joint placement of UAVs and radio resource allocation carried out by a MADRL approach based on Deep Q-Network (DQN). The choice of this particular Deep-RL approach was motivated by its fast convergence and high stability. We consider the deployment of physical agents, corresponding with the UAV-BSs deployed in our scenario, which acquire observations of the surrounding environment and take actions in order to maximize a reward. Each physical agent consists of two virtual agents that will follow different policies: the first one (allocation model) is in charge of allocating the radio resources to the associated users, while the second one (placement model) tries to place the UAV-BS in order to accommodate the best possible number of users. An optimal placement of the UAV strictly depends on the maximum number of users that the UAV-BS can satisfy, hence, the two policies are highly correlated. This is the reason why we considered communication between the two virtual agents, by using the decision of radio resources allocation as one of the inputs for the placement policy. A visual representation of the explained framework can be found in Fig. 2.

The environment can be modeled as a Markov Decision Process (MDP), where the tuple of state, observation, action, and reward is separately defined for the two different policies:

1) States: at a certain timestep t the observation from the UAV-BS consists of:

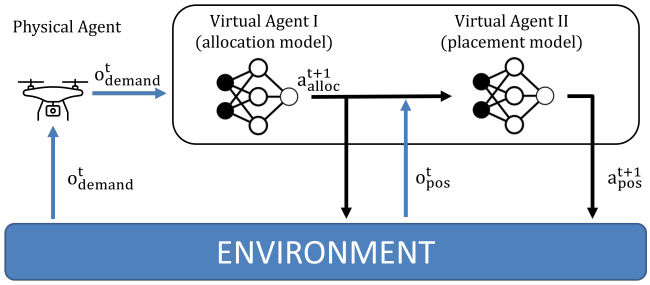


Fig. 2: Flowchart showing the information flow in our framework. The UAV-BS takes two observations from the environment. The first one, O^t_{demand}, is the input for the resource allocation strategy, while the second one, O^t_{pos}, together with the output from the allocation model, provides the input for the placement strategy.

- the aggregated data rate demand from all the associated users from a given slice s, agg_dem_s;
- the distance from all the users $d_{t,g}$;
- the relative angle from all the users $\alpha_{t,g}$;

The first observation is the input of the resource allocation policy, while the other two states, together with the output from the first policy, are the inputs for the placement strategy. Given the multi-agent peculiarity of this scenario, the agents will also get as input the relative position of the other UAV-BS, in order to share information regarding which area of the environment the others are currently covering.

2) Actions: each UAV-BS decides how to split the available bandwidth between the three slices, according to the users it is covering. This decision can be seen as a choice of three numbers, $\{bw_{eMBB}, bw_{URLLC}, bw_{mMTC}\} \in \mathbb{R}$ such that $bw_{eMBB} + bw_{URLLC} + bw_{mMTC} = 1$. The decision of the bandwidth allocation is part of the input for the placement policy, which will return five possible actions: straight, left, right, back, hover. These actions are turned into discrete movements of a given step-size that the UAV-BS will take at every decision step.
3) Reward: once each UAV-BS takes the actions regarding resource allocation and placement, they are rewarded according to the level of satisfaction achieved by the associated users. As previously mentioned, in this work we consider the SLA being based on the data rate only. Hence, for every UAV-BS, the reward that is returned at the end of a decision step will be the number of users associated to that base station ($G_{\text{conn}}(b)$) whose SLA is satisfied (Eq. 5), e.g., the number of users whose data rate is higher than a certain threshold th_g.

$$\delta_{b,t} = \sum_{g \in G_{\text{conn}}(b)} \mathbb{I}(dr_g >= th_g) \tag{5}$$

In our multi-agent configuration, each UAV-BS has two

DQNs, corresponding to its virtual agents, which act as the policies of radio resource allocation and placement. Each DQN consists of 3 feed-forward layers with ReLU activation function. The inputs of the networks are the states of the described MDP, while their outputs are the actions that the agents take. The rewards are used to train the neural networks according to the algorithm described in [15]. Every timestep, each agent collects an observation of the environment and feeds it to its DQNs. After the generated action is executed, the agent gets the immediate reward. The set of observation, action, reward, and next observation is stored in a memory, called replay buffer. When this dataset is sufficiently populated, it will be periodically sampled in order to train the agent's DQNs to maximize the cumulative reward over one episode. To make the training more robust to noise and speed up the learning convergence, we used a prioritized replay buffer [16], where the training batches are sampled from the buffer according to an importance weight associated to each of its entries.

III. Results

To assess the validity of our approach, we performed an extensive simulation campaign through multiple instances of the proposed scenario. The number of users and the size of the environment are not fixed, but vary according to the number of UAVs that are deployed. These parameters change in order to preserve the UAVs and users densities (Tab. II). The users are grouped in clusters, whose positions randomly change every new episode. The number of deployed UAVs is always smaller than the number of generated clusters, in order to make the placement problem more challenging. As the clusters of users, the UAVs are also randomly placed at the beginning of every new episode. Across the training procedure, the agents will learn how to properly cover the users starting from a random position, and how to fairly allocate radio resources to the slices in order to optimize the SLA satisfaction ratio of the users. Tab. II provides more details regarding the environment parameters, while information regarding the hyper-parameters of the training procedure can be found in Tab. III.

Fig.3 shows an example of user coverage with the highlighted trajectory followed by the agents. Furthermore, the green users represent the UEs whose SLA is currently satisfied.

We decided to compare our approach to three heuristics that have already been used in the literature before ([7], [8]):

- **Random**. Both UAVs' trajectories and radio resource allocations are randomly selected at every time step of the simulated scenario.
- **R**andom **A**llocation, **P**lacement on **C**entroids (RAPoC). Each UAV will decide one cluster to cover by positioning on its centroid. The radio resource allocation is randomly selected.
- **P**roportional **A**llocation, **P**lacement on **C**entroids (PAPoC). While each UAV will position on one of the centroids, the allocation will be proportional to the aggregated data rate demands (dem_{eMBB}, $\text{dem}_{\text{URLLC}}$, and dem_{mMTC}) of the users that are covered by the flying base station (Eq. 6).

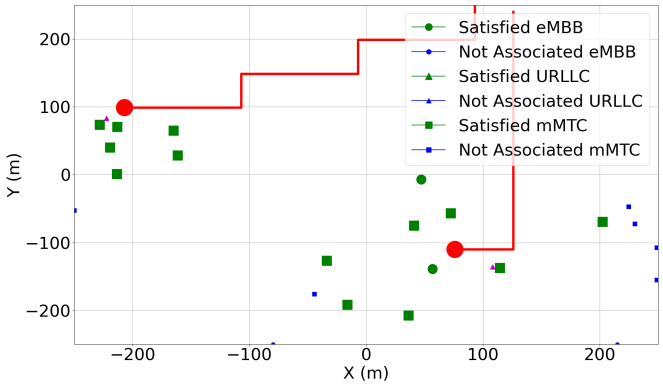


Fig. 3: Trajectory of 2 UAV-BSs (in red) following their own placement model. The green markers represent the satisfied users after the allocation of resources through our approach. Blue and purple markers respectively represent not associated and not satisfied users.

Parameter	Value
UAV transmit power p	1 W
pathloss exponend α	2.1 [13]
UAV half-power beamwidth η	30 deg. [13]
Near-field pathloss c	-38.4 dB [13]
Noise power σ^2	$8 \cdot 10^{-13}$ W
Bandwidth B	20 MHz
Users density, $\lvert\mathcal{G}\rvert/\text{km}^2$	100
UAVs number, $\lvert\mathcal{U}\rvert$	1 - 2 - 3 - 4 - 5
UAVs height h	50 m
UAVs movement step size	25 m
UAVs density $\lvert\mathcal{U}\rvert/\text{km}^2$	8
eMBB users demand dem_{eMBB}	5 Mbps [17]
URLLC users demand $\text{dem}_{\text{URLLC}}$	10 Mbps [17]
mMTC users demand dem_{mMTC}	0.5 Mbps [17]
p_{eMBB}, p_{URLLC}, p_{mMTC}	20%, 10%, 70%

TABLE II: Environment Parameters

$$\text{bw}_{\text{slice}} = \frac{\text{dem}_{\text{slice}}}{\text{dem}_{\text{eMBB}} + \text{dem}_{\text{URLLC}} + \text{dem}_{\text{mMTC}}} \tag{6}$$

We have compared our learning approach to the presented heuristics by measuring the average cumulative reward achieved by the whole set of users while exploiting the different policies. Fig. 5 shows the rewards achieved through the episodes in the scenario where two UAV-BSs are deployed. Furthermore, we show the average upper bound in the achieved reward, obtained by deploying a genie-aided solution obtained after solving to optimality the optimization model (3)-(4), which is possible if considering a perfect knowledge of the environment.

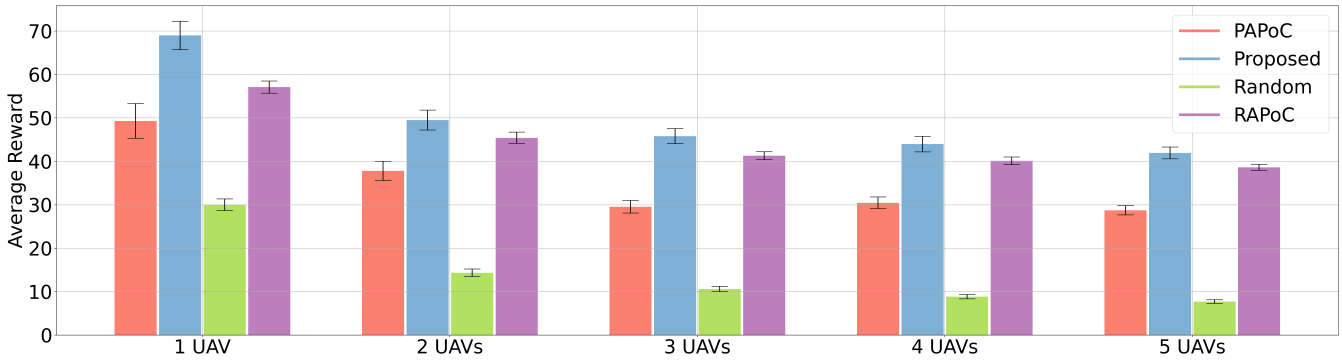


Fig. 4: Average reward and 95% confidence interval achieved with our proposed approach and compared to the other three heuristics by changing the number of UAV-BSs from 1 to 5.

Parameter	Value
Episodes per training	5000
Time-steps per episode	100
Episodes per testing	200
Discount factor	0.95
Learning rate	10^{-4}
Initial epsilon value	1
Epsilon decay value	0.99995
Minimum epsilon value	0.01
Replay memory size	1000
Batch size	32

TABLE III: Hyper-Parameters for Deep Q-Learning

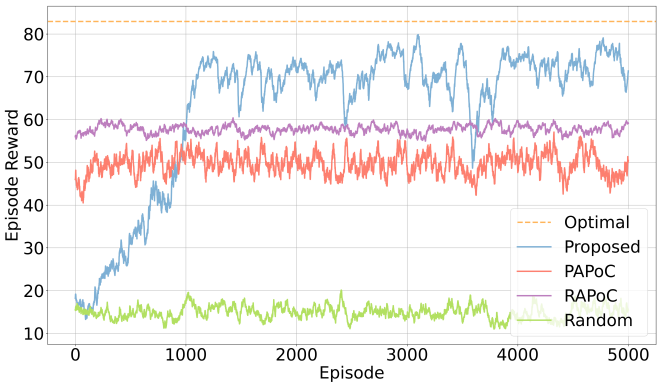


Fig. 5: Achieved rewards over episodes, comparing our proposed learning approach (blue line) to three heuristics: random (in green), RAPoC (in purple), and PAPoC (in salmon). The average optimal solution is also depicted in yellow.

At the beginning of the training process, the learning algorithm (blue line) performs similarly to the random heuristic (in red) because of the initial exploration phase carried out through an ϵ-greedy algorithm. After about 200 episodes, the policy starts to exploit the knowledge acquired, showing an increasing trend in the cumulative rewards that ends up outperforming the other heuristics after about 1000 episodes. The fluctuations present in the learned policy are due to the randomization of users' positions at the beginning of every new episode. This proves that, on average, our approach is also able to properly generalize, outperforming the results obtained from the presented heuristics in new environments that have not been observed before. In order to check the reliability of this algorithm to larger sets of agents, we tested the learning approach on a range of UAVs population spanning from 1 to 5. Fig. 4 presents the average and the 95% confidence interval of the cumulative rewards obtained by the agents over 200 episodes, each plot corresponding with a different fleet size.

From these latest results, it is possible to appreciate the robustness of our strategy with respect to a larger set of agents. The average reward will always outperform the other methods, with the correspondent confidence intervals that never overlap. Even in the most challenging scenario, when 5 UAVs are deployed causing a higher interference in the environment and a larger observation space, the learning approach is able to achieve a reward that is 10.5% higher than the best result obtained from the heuristics.

IV. Conclusion

In this work, we presented a multi-agent and multi-decision deep RL approach in a UAV-aided 5GNR network slicing scenario, with the aim of optimizing the SLA satisfaction ratio of the deployed users. We demonstrated how the learning approach can outperform heuristics that are commonly employed in the literature, by making smarter choices for the placement of UAVs and radio resource allocation for each slice. Furthermore, our learning strategy showed a good generalization capability, adapting the behavior of the agents according to the number of other agents and to different users' distributions. The promising results obtained throughout this research encourage us to investigate deeper into different directions:

1) partial observability of users' positions: the agents do not have a full understanding of the environment's state, but

will rely on partial observations that can be fostered by communication between base stations;
2) bigger environment and number of agents/users: a proper study on larger environments is needed to assess the scalability of the system with respect to larger environments and to more dense fleets;
3) different users' mobility model: in this work we focused on static UEs, while the presence of mobile users would deliver a more realistic and challenging scenario.

REFERENCES

[1] NGMN, "NGMN 5G white paper," *NGMN Alliance*, p. 125, 2015.
[2] S. Zhang, "An overview of network slicing for 5g," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.
[3] T. . 3GPP, "Enhancements of dedicated core networks selection mechanism (release 14)," September 2016.
[4] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on ran: Flexibility and resources abstraction," *IEEE Communications Magazine*, vol. 55, pp. 102–108, 06 2017.
[5] X. Foukas, M. Marina, and K. Kontovasilis, "Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture," 10 2017, pp. 127–140.
[6] M. A. Haque and V. Kirova, "5g network slice admission control using optimization and reinforcement learning," in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 854–859.
[7] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "Deepcog: Cognitive network management in sliced 5g networks with deep learning," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, 2019, pp. 280–288.
[8] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The lstm-based advantage actor-critic learning for resource management in network slicing with user mobility," *IEEE Communications Letters*, vol. 24, no. 9, pp. 2005–2009, 2020.
[9] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, "A tutorial on uavs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.
[10] P. Yang, X. Xi, K. Guo, T. Q. S. Quek, J. Chen, and X. Cao, "Proactive uav network slicing for urllc and mobile broadband service multiplexing," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 10, pp. 3225–3244, 2021.
[11] J.-W. Cho, P. Yang, T. Q. Quek, and J.-H. Kim, "Service-aware resource allocation design of uav ran slicing," in *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, 2020, pp. 801–805.
[12] M. Rumney, *LTE and the Evolution to 4G Wireless: Design and Measurement Challenges*, 2nd ed. Wiley Publishing, 2013.
[13] B. Galkin, B. Omoniwa, and I. Dusparic, "Multi-agent deep reinforcement learning for optimising energy efficiency of fixed-wing uav cellular access points," 2021.
[14] J. Fan, Q. Yin, G. Y. Li, B. Peng, and X. Zhu, "Mcs selection for throughput improvement in downlink lte systems," in *2011 Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, 2011, pp. 1–5.
[15] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: https://arxiv.org/abs/1312.5602
[16] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," 2015. [Online]. Available: https://arxiv.org/abs/1511.05952
[17] R. Li, Z. Zhao, Q. Sun, C.-L. I, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, 2018.