Spectrum-Aware Mobile Edge Computing for UAVs Using Reinforcement Learning

Babak Badnava
The University of Kansas
USA
babak.badnava@ku.edu

Taejoon Kim
The University of Kansas
USA
taejoonkim@ku.edu

Kenny Cheung
Universities Space Research Association (USRA)
USA
kcheung@usra.edu

Zaheer Ali
Universities Space Research Association (USRA)
USA
zali@usra.edu

Morteza Hashemi
The University of Kansas
USA
mhashemi@ku.edu

ABSTRACT

We consider the problem of task offloading by unmanned aerial vehicles (UAV) using mobile edge computing (MEC). In this context, each UAV makes a decision to offload the computation task to a more powerful MEC server (e.g., base station), or to perform the task locally. In this paper, we propose a spectrum-aware decision-making framework such that each agent can dynamically select one of the available channels for offloading. To this end, we develop a deep reinforcement learning (DRL) framework for the UAVs to select the channel for task offloading or perform the computation locally. In the numerical results based on deep Q-network, we consider a combination of energy consumption and task completion time as the reward. Simulation results based on low-band, mid-band, and high-band channels demonstrate that the DQN agents efficiently learn the environment and dynamically adjust their actions to maximize the long-term reward.

CCS CONCEPTS

• **Networks** → **In-network processing**; • **Computing methodologies** → *Distributed computing methodologies.*

KEYWORDS

Unmanned Aerial Vehicle (UAV), Mobile Edge Computing, Deep Reinforcement Learning.

ACM Reference Format:
Babak Badnava, Taejoon Kim, Kenny Cheung, Zaheer Ali, and Morteza Hashemi. 2021. Spectrum-Aware Mobile Edge Computing for UAVs Using Reinforcement Learning. In *The Sixth ACM/IEEE Symposium on Edge Computing (SEC '21), December 14–17, 2021, San Jose, CA, USA.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3453142.3491414

SEC '21, December 14–17, 2021, San Jose, CA, USA

ACM ISBN 978-1-4503-8390-5/21/12.
https://doi.org/10.1145/3453142.3491414

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have been experiencing healthy growth in the U.S. and around the world. For example, according to [Forecast 2020], the UAV market is expected to reach $21.8 billion by 2027. These UAVs can be used for various modern applications such as aerial photography, recreational flying, package delivery on a commercial basis, delivery of medical supplies, provision of support for search and rescue missions following natural calamities, and more. These use cases play an important role in assessing how to safely prepare for increased UAV usage moving forward.

There will be more application of UAS in the future and operation in visual line-of-sight (VLOS) from the Ground Control Station (GCS) will not provide a suitable environment for the same [Lin et al. 2018; Marojevic et al. 2019]. In fact, in order to truly unleash the potentials of UAS, real-world and commercial deployments will most likely be in the form of beyond visual line-of-sight (BVLOS) scenarios [Takacs et al. 2018], which in turn provide easier access to remote or hazardous areas, less human intervention, and reduced cost of operation. Beyond visual line-of-sight (BVLOS) operation will provide the assurance for the modern and futuristic applications of UAVs. Yet, compared with VLOS conditions, BVLOS carries higher safety risks since in the case of automated flights there may be no human observations, or the pilot may only be observing potential obstacles or other flying objects via a remote camera feed.

Supporting the safe and reliable operation of UAVs in BVLOS scenarios require two main functionalities: (i) real-time delivery of latency-sensitive command-and-control (C2) signals, and (ii) high-throughput payload data channel to the ground control station. According to [Kakar and Marojevic 2017], control and non-payload bandwidth requirements for unmanned aerial systems (UAS) in 2030 are projected to be 69.4 MHz and 39.5 MHz for a terrestrial communications infrastructure with and without video and weather radar data. On the other hand, payload data transfer typically requires much higher data rates and bandwidth and spectrum requirements.

To support spectrum needs for payload data transfer, we consider the application of mobile edge computing. In particular, we provide an architecture in which there are several UAVs connected to multiple base stations with edge computing capabilities. In this

scenario, the UAVs act as user equipment (UE) devices with constrained computation power and battery resources. We consider that there is a mission defined for each UAV, which requires extensive computing power. Due to limited on-board resources, the UAV nodes can offload some of their tasks to the base station (MEC server). We are interested in the problem of finding the optimal offloading policy that minimizes the completion time of all tasks while the lifetime of the system is also optimized. In this context, we define the lifetime of the system as the earliest time that at least one of the UAVs runs out of power. To solve this problem, we develop a deep reinforcement learning framework based on deep Q-network (DQN) agents, which observe the environment and take an action at any given time. A key contribution of this work is to provide a spectrum-aware edge computing framework for UAVs. In particular, our framework enables the UAVs to optimally select one of the available channels for offloading the computation tasks to one of the MEC servers.

The rest of this paper is organized as follows. Section 2 provides a brief background on reinforcement learning and related works. Section 3 presents the system model and problem formulation, followed by the proposed DRL framework in Section 4. In Section 5, we provide numerical results, and conclude the paper in Section 6.

2 BACKGROUND AND RELATED WORK

Reinforcement learning (RL) is a type of machine learning, in which an agent or a group of agents interact(s) with an environment by collecting **observations**, taking **actions**, and receiving **rewards**. The agent's experience is given by the tuple $(s_t, a_t, r_{t+1}, s_{t+1})$ such that at time step t, the agent observes the current state of the environment denoted by s_t, and chooses action a_t, which results in a reward $r_{t+1} = r(s_t, a_t, s_{t+1})$. Then, the state will transition to s_{t+1} according to the transition probability $p(s_{t+1}|s_t, a_t)$. The ultimate goal for the agent is to *learn* what is the optimal policy to maximize its cumulative reward over time.

Deep reinforcement learning has been proposed as an enhancement to more traditional RL approaches, and its applicability has been shown in multiple applications [Bagherpour et al. 2021; Ghasemi et al. 2020]. In a deep RL architecture, an agent uses a deep neural network as a function approximator to represent its policy and/or value function. This enables the observation space (and potentially the action space) to be continuous and uncountable. Deep Q-Network (DQN) [Mnih et al. 2013] is a specific deep RL agent, where its state-action value function is updated by minimizing the following loss function:

$$L(\theta) = \mathbb{E}\left[Q(s_t, a_t; \theta) - \left(r(s_t, a_t) + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta)\right)\right],$$

where $Q(s_t, a_t; \theta)$ represents the estimated value function for state s_t and action a_t and the set of DQN parameters denoted by θ. The loss function $L(\theta)$ is a direct result of the *Bellman Equation*.

Related Work: The authors in [Callegaro and Levorato 2021] proposed a framework in which they employ dynamic programming and RL to make optimal decisions between local and edge-assisted computing. They have successfully reduced the delay in task completion by using their proposed method, however, they did not consider the energy consumption of the UAVs.

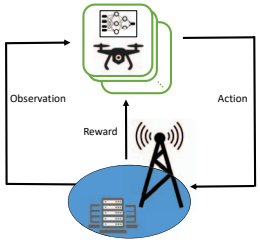


Figure 1: Deep reinforcement learning model.

The authors in [Zhou et al. 2018] considered a case where the network is equipped with wireless power transfer technologies and assumed that each UAV has access to unlimited energy. The UAVs are also able to transfer energy to the users, and users can use the harvested energy to perform local computation or offload their tasks to the MEC servers. They used sequential convex optimization techniques to optimize the number of offloading computation bits, the local computation frequencies of users and the UAV, and the trajectory of the UAV. The authors in [Chang et al. 2021] also assumed that the UAVs are computationally powerful and able to move from one user to another to help them with the completion of their tasks. They proposed a framework in which they optimize the quality of experience with flight path trajectory.

The model presented in [You and Huang 2016] consists of multiple users that have different computing tasks, each of which can be partially done by the user and the rest offloaded to a MEC server. In [You and Huang 2016], the authors formulated an optimization problem to minimize the total weighted energy consumption (local computing plus offloading to the base station), subject to a total fixed delay constraint and computation capacity constraint at all users and base station.

In [Yang et al. 2018], the authors investigated the problem of total energy minimization in an offloading setting. Their optimization model takes into account the transmission power as well as the local energy consumption for local execution. They considered a group consisting of two users and optimized the fraction of data to be offloaded by each user and the fraction of time allocated to each group of users. However, the base station is assumed to have perfect knowledge of channel state information as well as local computation capabilities and input data sizes across all users.

Most of the previous works considered the UAVs as an additional resource that can be used to improve the MEC and user's experience. However, there is a limited body of work that considers the UAVs as a UE that needs computational power to perform a task. In this paper, we consider the case where UAVs have a limited power but have access to different spectrum bands. Hence, they need to optimize the energy usage by choosing an optimal channel to offload their computation tasks to a central server.

3 SYSTEM MODEL

We consider a network with N MEC servers and K UAVs. Each UAV is equipped with C different radio, and can communicate with MEC servers through different channels. The area covered by the network is a $R \times R$ square. The MEC servers and UAVs are located randomly in this area, and the network operates in a time-slotted fashion, where the duration of each time interval is denoted by T. Each UAV is tasked to perform a mission with high computations

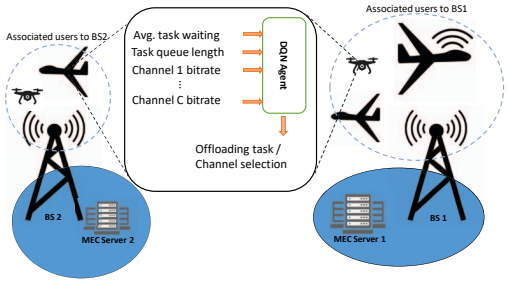


Figure 2: System model.

(e.g., target detection and tracking). Each mission is composed of different sub-tasks that would appear over time. The UAVs can either compute the tasks locally or offload the tasks to one of the MEC servers. In case of offloading to a MEC server, the UAV needs to communicate with a MEC server using one of the C communication channels available to it.

Mission and Task Arrival: Each mission is composed of multiple tasks that need to be completed over time. At a time instant t, each UAV $j \in \{1, ..., K\}$ needs to perform a set of computation tasks denoted by $\mathcal{M}_j(t)$. We assume each mission has a different task arrival rate λ_j, and model the number of incoming tasks at time t using a *Poisson* distribution with rate λ_j; i.e.,

$$|\mathcal{M}_j(t)| \sim \text{Pois}(\lambda_j), \forall j \in \{1, ..., K\}, \forall t. \tag{1}$$

The tasks, after arrival, are buffered in a queue and served on a first-in-first-out basis. The size of tasks, which is in bits, follows a uniform distribution in interval $(0, B_{\max})$.

Energy Consumption: At the beginning of the mission, all UAVs start from a full energy level of $E_{\max}$, and then gradually consume energy over time until depletion, in which case the system's lifetime is over. We use $E_i(t)$ to denote the energy level of the user i at the beginning of the time interval t. At each interval, each user either stays idle, does local computation of tasks, or offloads some tasks to its serving MEC server using one of the available channels. Denoting the action taken by the user j at time interval t by $a_{j,t}$, the energy level of the user evolves over time as follows:

$$E_j(t+1) = \begin{cases} E_j(t) - \epsilon & \text{if } a_{j,t} = \text{idle}, \\ E_j(t) - E_{j,\text{local}}(t) - \epsilon & \text{if } a_{j,t} = \text{local comp.}, \\ E_j(t) - E_{j,\text{offload}}(t) - \epsilon & \text{if } a_{j,t} = \text{offloading}. \end{cases}$$

where ϵ denotes the unit stand-by energy consumption for each user at every time interval.

System Lifetime: We assume that the system with K UAVs crashes once at least one of them runs out of power. The time that the first UAV depletes its battery leads to the definition of the *system lifetime*, denoted by LT, as follows:

$$\text{LT} = \max\left\{t \mid E_j(t) > 0, \forall j \in \{1, ..., K\}\right\}. \tag{2}$$

Having defined these metrics, our goal is to maximize the system lifetime as much as possible.

Computation Model: As mentioned above, each UAV can either compute its incoming tasks using its local processor or offload it to a MEC server. For the local computation, We adopt a similar model to [Chen and Wang 2018; Naderializadeh and Hashemi 2019], where the UAV first computes its maximum feasible computing power at any interval, and uses that to compute the bit budget. To be precise, for user $j \in \{1, ..., K\}$, the maximum power spent for local computing at time interval t is calculated as $P_j^{\max}(t) = \frac{E_j(t)}{T}$. Then, the maximum feasible CPU frequency is computed as:

$$f_{j,\text{local}}^{\max}(t) = \min\left\{f_{j,\text{local}}^{\max}, \sqrt[3]{\frac{P_j(t)}{\kappa}}\right\}, \tag{3}$$

where $f_{j,\text{local}}^{\max}$ denotes the absolute maximum CPU frequency, and κ represents the effective switched capacitance. Then, the maximum number of bits that can be computed by user j at time t is given by:

$$B_{j,\text{local}}^{\max}(t) = \left\lfloor \frac{T \times f_{j,\text{local}}^{\max}(t)}{L_j} \right\rfloor, \tag{4}$$

where L_j denotes the number of CPU cycles per bit at user j. The user then checks its task buffer, and computes the tasks at the head of the queue one by one, as long as the total number of computed bits does not exceed $B_{j,\text{local}}^{\max}(t)$. Note that if the size of the first task is already larger than $B_{j,\text{local}}^{\max}(t)$, then the user remains idle and does not do any local computation at that step. We denote the effective consumed energy for the local computation of user j at time interval t by $E_{j,\text{local}}(t)$.

Communication Model: The UAVs also can compute their incoming tasks by offloading them to a nearby MEC server. We assume each UAV, at each time instance, is associated with a MEC server that has the strongest long-term channel gain to it. We denote by S_j the MEC server to whom the UAV j is associated. However, since we assume that each UAV is equipped with C different radios, the association is done for each channel.

For the UAV j to offload its computation tasks to the server S_j at a time interval t, it first calculates its maximum feasible transmit power (i.e., $P_j^{\max}(t)$) based on its instantaneous energy level. Then, it uses the maximum feasible power to send the computation task to the MEC server. The maximum uplink achievable rate is:

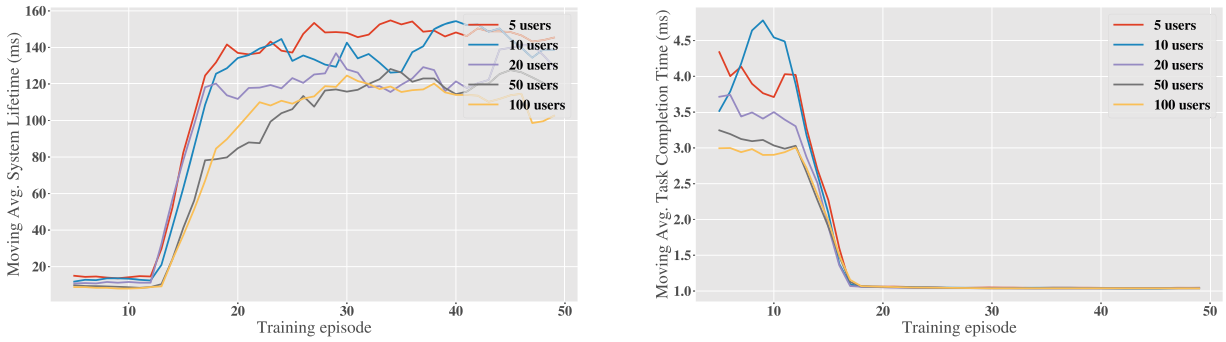
$$R_j^{\max}(t) = W_{S_j,j}(t) \log_2\left(1 + \beta_{S_j,j}(t) \min\left[P_j^{\max}(t), P_{j,\text{Tx}}^{\max}\right]\right),$$

where $W_{S_j,j}(t)$ denotes the amount of bandwidth allocated to the uplink transmission between user j and server S_j at time interval t. Note that the bandwidth of different channels are different, and the DQN agent dynamically selects the optimal channel. Moreover, $\beta_{S_j,j}(t)$ denotes the received signal-to-noise ratio (SNR) from user j to server S_j, which depends on the selected channel. In this case, $P_{j,\text{Tx}}^{\max}$ is the maximum transmit power of user j. The uplink transmissions of users to their respective MEC servers at each time interval may share the spectrum using multiple access techniques, such as FDMA or TDMA. Therefore, the maximum number of bits that user j can transmit to server S_j at time t is obtained as:

$$B_{j,\text{offload}}^{\max}(t) = \left\lfloor T \times R_j^{\max}(t) \right\rfloor. \tag{5}$$

Similar to local computation, the user offloads tasks from head of its task buffer whose total number of bits does not exceed $B_{j,\text{offload}}^{\max}(t)$. We denote the effective consumed energy by user j to offload its tasks to its associated server at time interval t by $E_{j,\text{offload}}(t)$.

Channel Model: As mentioned before, each UAV is equipped with C radio across different spectrum bands. Each channel has a specific center frequency and bandwidth. We consider an urban



(a) (b)

Figure 3: (a) System Lifetime learning curve (b) Task Completion learning curve. The DQN agent learns the optimal policy for both objectives (increased system lifetime and decreased task completion time) almost after 25 episodes of learning.

macro environment, and adopt the path loss model presented in [38.901 2020], Table 7.4.1-1. In this model, the line of sight (LOS) pathloss is calculated as follows:

$$PL_{UMa-LOS} = \begin{cases} PL_1 & \text{if } 10 \le d_{2D} \le d_{BP}, \\ PL_2 & \text{if } d_{BP} \le d_{2D} \le 5km, \end{cases}$$

where $d_{BP} = 4h_{BS}h_{UE}f_c/c$ such that h_{BS} and h_{UE} are effective antenna heights at the UE and BS, f_c is the carrier center frequency, and $c = 3 \times 10^8 m/s$. In this channel model, d_{2D} and d_{3D} are the 2D and 3D distance between UE and BS, respectively. Also, we have:

$$PL_1 = 28 + 22\log_{10}(d_{3D}) + 20\log_{10}(f_c),$$

$$PL_2 = 28 + 40\log_{10}(d_{3D}) + 20\log_{10}(f_c) - 9\log_{10}(d_{BP}^2 + h_{UT}^2).$$

Furthermore, we obtain the Non-LOS path loss as follows:

$$PL_{UMa-NLOS} = \max(PL_{UMa-LOS}, PL'_{UMa-NLOS}),$$

where

$$PL'_{UMa-NLOS} = 13.54 + 39.08\log_{10}(d_{3D}) + 20\log_{10}(f_c) - 0.6(h_{UE} - 1.5).$$

Our channel model also incorporates the Rayleigh short term fading, as presented in [Li and Huang 2002].

4 PROPOSED DEEP REINFORCEMENT LEARNING APPROACH

In order to choose the best communication channel, we propose to employ a DQN agent, as shown in Figure 2, with each UAV. In this case, the DQN agent decides whether it is best to stay idle, perform the task locally, or offload it to one of the MEC servers. The DQN agent also decides, in case of offloading, to choose which communication channel.

Observations and Actions We consider an episodic scenario, where at the beginning of each time step, the UAV is able to observe its energy level, average waiting time for tasks, task queue length, and channel bit rate for all channels. Then, the DQN agent decides which channel results in a lower energy consumption, and selects that channel to offload computation tasks to one of the MEC servers.

Rewards: As mentioned in Section 3, our goal is to maximize the system lifetime and amount of completed tasks (i.e., minimum completion time). The system lifetime is a function of users' energy level. Hence, minimizing the amount of energy consumed by each user at each time step leads to an improved system lifetime. The amount of consumed energy at each user is available at the user-side and the DQN can use this measure to maximize the system lifetime, but since RL algorithms are reward maximization problems by design, we define the reward function as the number of computed bits divided by the consumed energy, i.e.,

$$r_{t+1} = \frac{b_j(t)}{E_j(t) - E_j(t+1)},$$

where $b_j(t)$ is the number of completed bits corresponds to mission of UAV j at time t.

5 NUMERICAL RESULTS

In this section, we present our simulation results to demonstrate the efficacy of the proposed method. We consider a network area of size $1000m \times 1000m$. Each UAV is connected to this network through four different channels with frequencies and bandwidths presented in Table 1. We assume a time interval length of $T = 100ms$, and at the beginning of each episode, the energy level of each user is set to $1J$. Each UAV communicates with the base stations at the maximum transmission power of 27 dBm, and the channel's noise variance is set to be −174 dBm/Hz.

Band ID	Center Frequency	Bandwidth (MHz)
1	600 MHz	10
2	850 MHz	20
3	2.4 GHz	40
4	5.9 GHz	100

Table 1: Center frequency and bandwidth of all available channels. Given the current state, the DQN agent of each UAV optimally selects a channel at each time slot.

We assume that all the UAVs are performing the same mission. Hence, they have the same task arrival rate. Thus, we set the mean task arrival rate to be 10, the task length to be 1 KB and the unit stand-by energy is set to $\epsilon = 10^{-7}$. The CPU frequency of servers

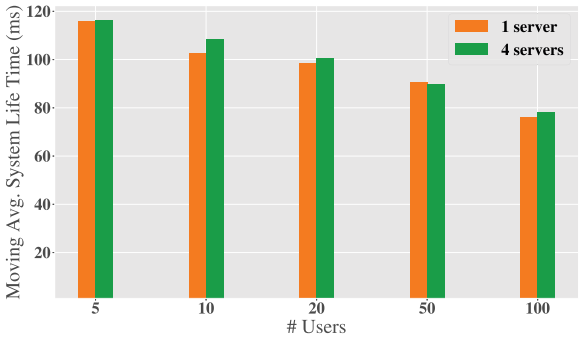


Figure 4: System Lifetime comparison for different number of MEC severs as the number of UAVs increases. The system lifetime decreases as the number of UAVs increases.

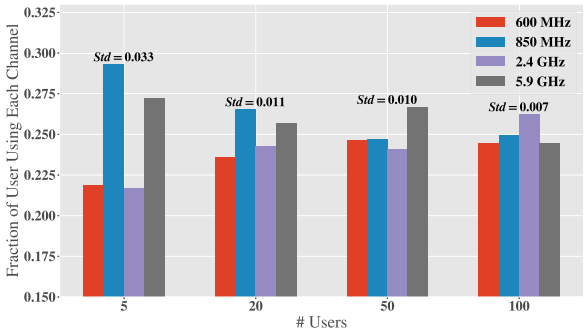


Figure 5: Channel selection distribution for different number of users. As the number of users increases, the DQN allocate the channel to users more uniformly.

and UAVs are set to be 3 GHz and 1 GHz, respectively, with respective cycles per bit of 1000 and 500. The effective switched capacitance is set to $\kappa = 10^{-27}$.

Similar to [Naderializadeh and Hashemi 2019], the DQN agent is equipped with a 2-layer neural network with 32 nodes per layer and *tanh* activation function for the hidden layers. The discount factor of the DQN agent is set to $\gamma = 0.9$, and it uses a *softmax* policy during the training process. The parameters of the neural networks are updated at the end of each episode by taking a batch of 64 random samples from the buffer. The learning rate also starts at 5×10^{-3} and is cut in half every 100 episodes.

Figure 3 shows the moving average of the system lifetime during the training process of the DQN agent. From the results, we note that the implemented DQN agents converge after around 25 episodes of learning. Moreover, Figure 4 shows that the average system lifetime decreases as the number of UAVs increases, as expected. Also, the overall system lifetime slightly improves with 4 MEC servers. Finally, Figure 5 presents the fraction of UAVs that select a specific channel. From the result, one can notice that as the number of users increases, the DQN agents learn to uniformly select the channels. In this way, channel congestions are avoided. We note that our preliminary results on channel selection can be extended in several directions by, for example, incorporating more realistic spectrum sensing and sharing models.

6 CONCLUSION

In this paper, we considered a network of UAVs such that each of them is performing a computationally heavy mission. This network is equipped with multiple MEC servers as well, and the UAVs can communicate with the MEC servers through different channels. In order to save their energy for mission completion, they need to learn which channel provides a higher bit rate with a given transmit power. Our proposed DRL framework based on DQN agents enable each UAV to observe the channel bit rate and UAV's task queue status, and then makes a decision to offload its task to a MEC server using one of the channels or locally perform the task. We evaluated the performance of our method through simulation, and our numerical results demonstrate the effectiveness of the proposed method in increasing the system's lifetime.

ACKNOWLEDGMENTS

This work was partially supported by a NASA/USRA grant, and NSF grants CNS-1948511 and CNS-1955561.

REFERENCES

TS 38.901. 2020. *5G; Study on channel model for frequencies from 0.5 to 100 GHz (3GPP TR 38.901 version 16.1.0 Release 16).* 3GPP TSG RAN: V16.1.0.

Reza Bagherpour, Nasser Mozayani, and Babak Badnava. 2021. Improving demand-response scheme in smart grids using reinforcement learning. *International Journal of Energy Research* (2021).

Davide Callegaro and Marco Levorato. 2021. Optimal Edge Computing for Infrastructure-Assisted UAV Systems. *IEEE Transactions on Vehicular Technology* (2021).

Huan Chang, Yicheng Chen, Baochang Zhang, and David S. Doermann. 2021. Multi-UAV Mobile Edge Computing and Path Planning Platform based on Reinforcement Learning. *arXiv:abs/2102.02078* (2021).

Zhao Chen and Xiaodong Wang. 2018. Decentralized Computation Offloading for Multi-User Mobile Edge Computing: A Deep Reinforcement Learning Approach. *arXiv:1812.07394* (2018).

Global Forecast. 2020. Unmanned Aerial Vehicle (UAV) Market by Component, Class, End User, Type, Capacity, and Mode of Operation. https://www.meticulousresearch.com/product/unmanned-aerial-vehicle-UAV-market-5086. [Online].

Arman Ghasemi, Amin Shojaeighadikolaei, Kailani Jones, Morteza Hashemi, Alexandru G. Bardas, and Reza Ahmadi. 2020. A Multi-Agent Deep Reinforcement Learning Approach for a Distributed Energy Marketplace in Smart Grids. In *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm).*

Jaber Kakar and Vuk Marojevic. 2017. Waveform and spectrum management for unmanned aerial systems beyond 2025. In *IEEE 28th Annual international symposium on personal, indoor, and mobile radio communications (PIMRC).*

Yunxin Li and Xiaojing Huang. 2002. The simulation of independent Rayleigh faders. *IEEE Transactions on Communications* (2002).

Xingqin Lin, Vijaya Yajnanarayana, Siva D. Muruganathan, Shiwei Gao, Henrik Asplund, Helka-Liina Maattanen, Mattias Bergstrom, Sebastian Euler, and Y.-P. Eric Wang. 2018. The Sky Is Not the Limit: LTE for Unmanned Aerial Vehicles. *IEEE Communications Magazine* (2018).

Vuk Marojevic, Ismail Guvenc, Mihail Sichitiu, and Rudra Dutta. 2019. An Experimental Research Platform Architecture for UAS Communications and Networking. In *IEEE 90th Vehicular Technology Conference (VTC2019-Fall).*

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning. *arXiv:1312.5602* (2013).

Navid Naderializadeh and Morteza Hashemi. 2019. Energy-aware multi-server mobile edge computing: A deep reinforcement learning approach. In *53rd Asilomar Conference on Signals, Systems, and Computers.*

Attila Takacs, Xingqin Lin, Stephen Hayes, and Erika Tejedor. 2018. Drones and networks: Ensuring safe and secure operations. In *Ericsson, Tech. Rep.*

Zhaohui Yang, Jiancao Hou, and Mohammad Shikh-Bahaei. 2018. Energy Efficient Resource Allocation for Mobile-Edge Computation Networks with NOMA. In *IEEE Globecom Workshops (GC Wkshps).*

Changsheng You and Kaibin Huang. 2016. Multiuser Resource Allocation for Mobile-Edge Computation Offloading. In *IEEE Global Communications Conference (GLOBECOM).*

Fuhui Zhou, Yongpeng Wu, Haijian Sun, and Zheng Chu. 2018. UAV-Enabled Mobile Edge Computing: Offloading Optimization and Trajectory Design. In *IEEE International Conference on Communications (ICC).*