Learning Based Edge Computing in Air-to-Air Communication Network

Zhe Wang
Dept. of Electrical and Computer Engineering
University of Louisville
Louisville, United States
zhe.wang@louisville.edu

Hongxiang Li
Dept. of Electrical and Computer Engineering
University of Louisville
Louisville, United States
h.li@louisville.edu

Eric J. Knoblock
Communications and Intelligent Systems Division
NASA Glenn Research Center
Cleveland, OH, United States
eric.j.knoblock@nasa.gov

Rafael D.Apaza
Communications and Intelligent Systems Division
NASA Glenn Research Center
Cleveland, OH, United States
rafael.d.apaza@nasa.gov

***Abstract*—This paper studies learning-based edge computing and communication in a dynamic Air-to-Air Ad-hoc Network (AAAN). Due to spectrum scarcity, we assume the number of Air-to-Air (A2A) communication links is greater than that of the available frequency channels, such that some communication links have to share the same channel, causing co-channel interference. We formulate the joint channel selection and power control optimization problem to maximize the aggregate spectrum utilization efficiency under resource and fairness constraints. A distributed deep Q learning-based edge computing and communication algorithm is proposed to find the optimal solution. In particular, we design two different neural network structures and each communication link can converge to the optimal operation by exploiting only the local information from its neighbors, making it scalable to large networks. Finally, experimental results demonstrate the effectiveness of the proposed solution in various AAAN scenarios.**

***Index Terms*—AAAN, deep Q-learning, edge computing, resource allocation**

I. Introduction

As the airspace is becoming more crowded and complex, Air-to-Air (A2A) communications are becoming increasingly important in both military and civil aviation applications. In particular, the National Aeronautics and Space Administration (NASA) is investigating advanced A2A communication technologies to facilitate the Urban Air Mobility (UAM) and Advanced Air Mobility (AAM) concepts and help emerging aviation markets to safely develop an air transportation system that moves people and cargo between places previously not served or underserved by aviation – using new aircraft that are only just now becoming possible [1]. As in many other wireless communication systems, spectrum scarcity stands out as the main challenge in aeronautical communications. To address this challenge, the concept of autonomous spectrum management was developed for A2A and Air-to-Ground(A2G) communications to modernize the existing Air Traffic Control (ATC) system and conceive future UAM/AAM applications [2].

As artificial intelligence and machine learning are taking center stage, learning-based technologies have drawn significant attention in many fields including aeronautical communication and networking [3]. Compared to conventional model-based approaches, well-trained neural networks can make quick decisions in complex environments that are intractable in practice. On the other hand, to reduce communication latency and save bandwidth, edge computing has recently emerged as a distributed computing paradigm that brings computation and data storage closer to the end user [4]. Under the context of aeronautical communications, edge computing enables quick decision making by each operating aircraft, which is in contrast to existing aviation systems relying on centralized ground control. In particular, edge computing-based A2A communications allow aircraft within the communication range to directly exchange information without having to route the data through the ground infrastructure.

In this paper, we consider a distributed Air-to-Air Ad-hoc Network (AAAN), which is featured by high mobility, lack of central control, and self-organization. The objective is to maximize the aggregate spectrum utilization efficiency under resource and fairness constraints. In particular, we propose a distributive optimization framework that leverages edge computing and deep reinforcement learning for real-time decision making in dynamic network environments.

Deep Reinforcement Learning (DRL) has been recently applied to solve dynamic spectrum access problems [5] and power control problems [6], [7] in various wireless networks. In [5], Naparstek and Cohen proposed a deep multi-user reinforcement learning algorithm for distributed dynamic spectrum access, where the long short-term memory model is adopted to extract essential features for spectrum prediction. Along a different line, the authors in [6] proposed a DRL-based algorithm for power allocation, without considering the channel selection problem. Additionally, Liang et. al. proposed joint spectrum and power allocation algorithms for

vehicular communications with delayed CSI feedback [7]. However, these studies are limited to Device-to-Device (D2D) communications or vehicular networks, which rely on cellular networks as the underlying infrastructure.

Relevant studies on aerial edge computing mostly involve A2G communications. In [8], a new Mobile Edge Computing (MEC) framework was proposed from an A2G integration perspective, where a case study is conducted to demonstrate the performance improvements in computation capability and communication connectivity based on real-world road topology. Moreover, the authors in [9] applied MEC to minimize the energy optimization for A2G integrated wireless networks. Recently, a survey on A2G integrated mobile edge networks was conducted in [10]. Other studies on edge computing for D2D communications in the mobile cellular network can be found in [11]–[13].

Different from those terrestrial or A2G networks, we consider a generic AAAN that is completely distributive without ground control. Optimization of such a network poses multiple challenges. First, due to each aircraft's high mobility, the network is highly dynamic and a communication link suffers from transient connection and disconnection. Second, the flight safety information is time-critical and must be delivered in real time. Third, multiple communication links must compete and share a limited spectrum to balance the aggregate network throughput and fairness among individual aircraft. To tackle these challenges, we propose a Learning based Edge Computing Resource Allocation (LECRA) algorithm for A2A communications. This paper extends our previous work in [14] to meet the aircraft's individual quality of service (QoS) requirements. Our contributions are summarized as follows:

- LECRA is flexible to balance the spectrum utilization efficiency and user fairness, where the QoS associated with different communication links is adjustable to reflect their unequal priorities or importance.
- LECRA adopts two alternative neural network structures based on Deep Q-learning, which is a model-free and off-policy algorithm. It is robust to produce the optimal channel and power allocation solutions, even in a highly dynamic aerial environment.
- LECRA is completely distributive, where each A2A communication link only needs to obtain local information from their neighbors to make decisions. Therefore, LECRA enjoys high scalability by taking advantage of edge computing and communication.

The rest of the paper is organized as follows. Sec. II describes the system model and formulates the joint resource allocation problem. The LECRA algorithm is discussed with details in Sec. III. Experimental results are presented in Sec. IV to evaluate the performance of the LECRA algorithm in different scenarios. Finally, a conclusion is drawn in Sec. V.

II. System Model and Problem Formulation

In this paper, we consider a single-hop point-to-point AAAN communication network, as illustrated in Fig. 1. We assume a total of M aircraft $\{A_1, A_2, ..., A_M\}$, each equipped

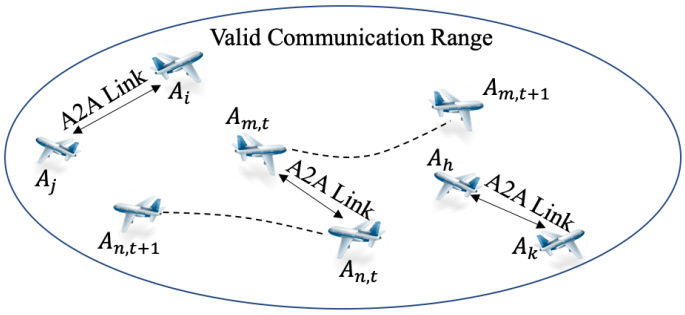


Fig. 1: Single Hop A2A Communications.

with a single antenna, and N pair of A2A communication links $\{L_1, L_2, ..., L_N\}$. In AAAN communications, the aircraft are constantly moving and the surrounding environment is constantly changing, so the spectrum access and power control must be updated in a dynamic fashion. Two aircraft are defined as *neighbors* if they are within the direct communication range R, and only neighbors can establish a communication link. For example, Fig. 1 shows three aircraft pairs (A_i, A_j), (A_m, A_n), and (A_h, A_k) establish three communication links at time t. At time $t+1$, aircraft $A_{m,t+1}$ and $A_{n,t+1}$ move out of their direct communication range and become disconnected. Meanwhile, aircraft $A_{m,t+1}$ and $A_{n,t+1}$ respectively become neighbors of $A_{h,t+1}$ and $A_{j,t+1}$, so new communication links can be established.

Due to spectrum scarcity, we consider a spectrum limited communication scenario where the number of available frequency channels, K, is always less than the number of A2A links (i.e., $K < N$). As a result, some communication links have to simultaneously share the same channel, causing co-channel interference. For fairness consideration, we impose that no communication link can use more than one channel at any given time [15] and a minimum QoS shall be maintained for each link. In this case, the Signal to Interference plus Noise power Ratio (SINR) for link n on channel k is given by

$$SINR_{n,k} = \frac{\phi_{n,k} p_{n,k} |h_{n,k}|^2}{\sigma^2 + \sum_{v \in \mathcal{V}} \phi_{v,k} p_{v,k} |h_{v \to n,k}|^2} \tag{1}$$

where $\phi_{n,k}$ is the channel selection indicator at time t (i.e., $\phi_{n,k} = 1$ when channel k is selected by link n, otherwise $\phi_{n,k} = 0$, and $\sum_{k \in \mathcal{K}} \phi_{n,k} \leq 1$); $h_{n,k}$ and $p_{n,k}$ denote link n's channel gain and transmit power on channel k, and σ^2 represents the AWGN power. Suppose link n is interfered by a set of neighboring links $\mathcal{V}$ transmitting on the same channel, $h_{v \to n,k}$ denotes the channel gain of the interference link.

To simplify the problem, we assume unit bandwidth so that the spectrum efficiency becomes the data rate, which is $R_{n,k} = log_2(1 + SINR_{n,k})$. In order to maximize the sum rate, the joint channel selection and power control problem is

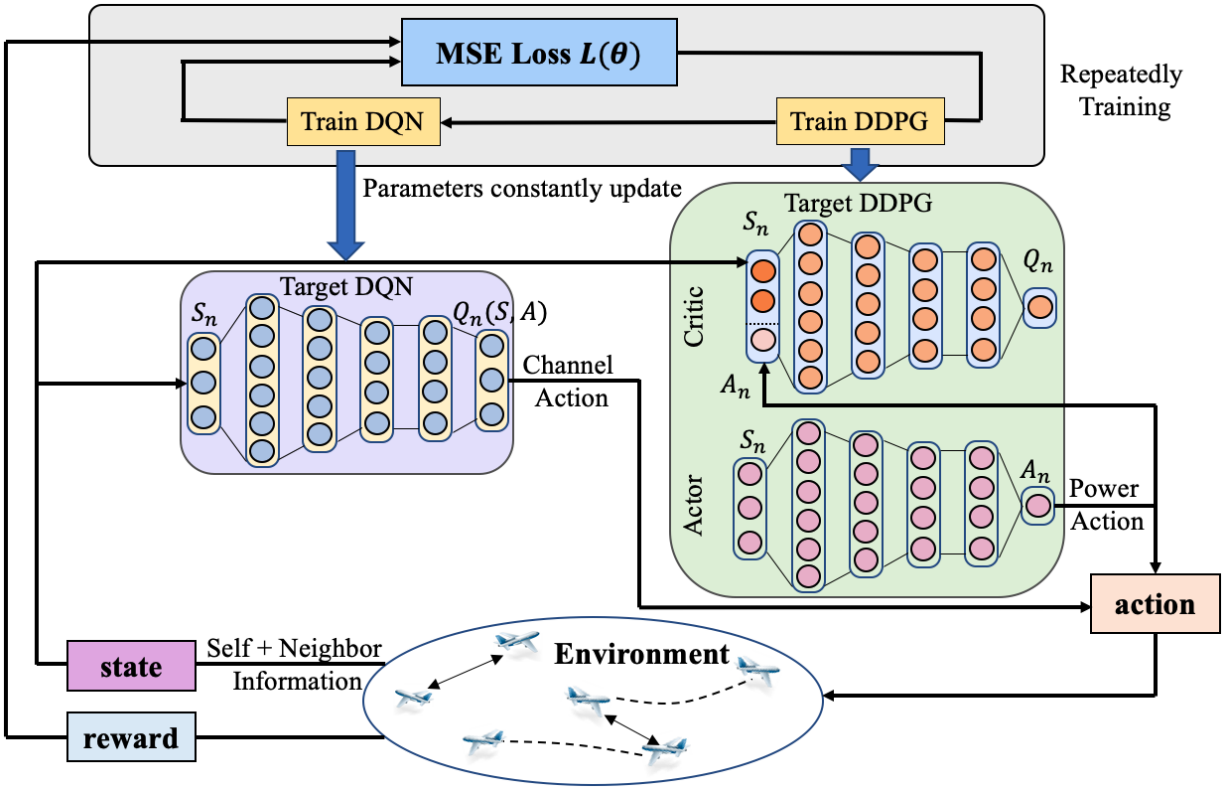


Fig. 2: DRL Training Model in LECRA

formulated as follows:

$$\begin{aligned} &\max_{\phi,p} \sum_{n=1}^{N}\sum_{k=1}^{K} R_{n,k} \\ s.t. \quad &0 \le p_{n,k} \le P_{max}, \forall L_n \in \mathcal{L} \\ &\sum_{k=1}^{K} \phi_{n,k} \le 1, \forall L_n \in \mathcal{L}, \forall k \in \mathcal{K} \\ &SINR_n \ge SINR_n^{min} \end{aligned} \tag{2}$$

where channel selection $\phi_{n,k} \in \{0,1\}$ and transmit power $p_{n,k} \in [0, P_{max}]$ are optimization variables, and $SINR_n^{min}$ is the minimum QoS required by link n, representing user fairness.

To solve the optimization problem in (2), we propose a DRL-based joint resource allocation algorithm that can be implemented in a distributive manner via edge computing and communication. As shown in Fig. 2, each communication aircraft is a learning agent equipped with a decision engine to solve the problem (2) using its local information. We assume that each aircraft can exchange information with its neighbors through a common control channel. More specifically, each communication link can obtain information about its own link and its neighboring links. Based on the premise that the dynamic network and communication patterns are correlated in time, it is reasonable to expect the A2A pairs can learn from their historical data and optimize their current decisions on channel selection and power allocation.

III. Learning-based Joint Resource Allocation

Our LECRA algorithm is based on deep Q-learning, which is an off-policy deep reinforcement learning algorithm to solve dynamic programming problems. It has the advantages of evaluating the expected utility among available actions without prior knowledge of the system model, as well as handling stochastic transitions without adaptations. Fig. 2 illustrates the DRL training model in LECRA, where each communication link makes its own decisions. Specifically, the goal of each agent is to learn an optimal policy that maximizes the accumulated reward in an observable Markovian environment. The agent must find out which action yields the most reward through trial and error rather than explicit instructions. This can be achieved by successively improving its Q value of particular actions at particular states, where each state-action pair has a particular Q value and all of them are stored in a Q table.

LECRA consists of several major elements: action, state, reward, transition, and policy. Given an arbitrary communication link, we use $\mathcal{V}$ and $\mathcal{D}$ to denote the sets of links causing/suffering interference to/from the chosen link respectively.

a) Action: In A2A communications, in every coherence time slot each agent needs to take two actions: channel selection and power assignment. We use $\mathcal{K}$ to represent channel action set, which consists of K actions because there are K channels available. From Section II, we know each agent can select no more than one channel during the same time slot

(i.e., $\sum_{k\in\mathcal{K}} \phi_{n,k}^{(t)} \leq 1, \forall \mathcal{L}$). To reduce the search space, the maximum power P_{max} is discretized to δ levels so that the power assignment action set is $\mathcal{P} = \{\frac{P_{max}}{\delta}, \frac{2P_{max}}{\delta}, ..., P_{max}\}$. Thus, the resource allocation action space is denoted by $\mathcal{A} = \{(\phi_k^{(t)}, p) | k \in \mathcal{K}, p \in \mathcal{P}\}$. For example, $a_n^{(t)} = \{\phi_{n,k}^{(t)}, p_{n,k}^{(t)}\}$ means that, at time t, agent n takes channel action $\phi_{n,k}^{(t)}$ and power action $p_{n,k}^{(t)}$. Accordingly, each agent has an action space of size $K\delta$.

b) Reward: Our distributed approach delegates the optimization problem in (2) to individual agents who locally maximize the sum rate. To this end, the reward of each agent is calculated as:

$$r_n^{(t+1)} = \sum_{k\in\mathcal{K}} R_{n,k}^{(t)} - \sum_{d\in\mathcal{D}^{(t+1)}} \lambda_{n\to d}^{(t)} - \lambda_{SINR}^{(t)} \quad (3)$$

where the first term is the data rate of the given link. In the second term, $\mathcal{D}^{(t+1)}$ denotes the neighboring link set that will be interfered by link n at time $t+1$, and penalty $\lambda_{n\to d}^{(t)}$ represents the sum rate loss in $\mathcal{D}^{(t+1)}$ due to link n's transmission:

$$\lambda_{n\to d}^{(t)} = \sum_{k\in\mathcal{K}} R_{d\backslash n,k}^{(t)} - \sum_{k\in\mathcal{K}} R_{d,k}^{(t)} \quad (4)$$

where $R_{d\backslash n,k}^{(t)}$ is the spectral efficiency of each link $d \in \mathcal{D}^{(t+1)}$ without considering the interference from link n:

$$R_{d\backslash n,k}^{(t)} = log_2(1 + \frac{\phi_{d,k}^{(t)} p_{d,k}^{(t)} |h_{d,k}^{(t)}|^2}{\sigma^2 + \sum_{x\neq n,d} \phi_{x,k}^{(t)} p_{x,k}^{(t)} |h_{x\to d,k}^{(t)}|^2}). \quad (5)$$

The last term of equation (3) is the penalty incurred when link n's estimated SINR is below the threshold $SINR_n^{min}$. It is worth noting that in practice, the agent calculates the reward at the beginning of time slot $t+1$ by using delayed information from time slot t.

c) State: The state vector s describes the status of the AAAN and it has the following elements: (i) Given link n's channel selection, transmitting power, channel gain, received interference-plus-noise power, and data rate. (ii) $\mathcal{V}$ link's channel selection, transmitting power, channel gains, and date rate. (iii) $\mathcal{D}$ link's channel gains, data rate, and received interference-plus-noise power. To reduce the computation complexity and communication overhead, we set a threshold $N_{neighbor}$ to limit the number of neighboring links that can be included in the state vector, i.e., $|\mathcal{V}_n^{(t)}| = |\mathcal{D}_n^{(t)}| = N_{neighbor}$.

d) Policy: Given the state vector s, a policy $\pi(s, a)$ produces resource allocation action a. In LECRA, we design two alternative neural network structures (i.e., Deep Q-learning Network (DQN) and DQN plus Deep Deterministic Policy Gradient (DQN+DDPG)) to generate different learning policies. Specifically, DQN has a fully connected neural network that can predict discrete actions. On the other hand, DDPG is a model-free off-policy algorithm based on the deterministic policy gradient [16]. The DDPG structure includes a critic network and an actor-network, both of which are constructed by fully connected neural networks but have different functions. The critic network is used to learn critic, where the training process adopts the Bellman equation to find the optimal $Q(s, a)$. It is well known that DDPG can predict continuous actions, which is suitable to solve our continuous power allocation problem. Fig. 2 shows the DDPG+DQN training model, which becomes the DQN training model by removing the DDPG network.

Algorithm 1 Learning based Edge Computing Resource Allocation (LECRA)

```
Initialize Q value parameter θ_train with random values, and θ_target = θ_train.
Initialize replay memory E with zero value
Initialize state s
for t = 1, t_m do
    if t − 1 < B or rand(t) < ε then
        Randomly select an action a^(t) ∈ A
    else
        Action a^(t) = max_a Q(s^(t), a; θ_target)
    end if
    Perform action a^(t) on environment
    Get updated state s^(t+1)
    Calculate reward r^(t+1) using equation (3)
    Store e^(t) = (a^(t), s^(t), r^(t+1), s^(t+1)) in E
    if t − 1 > B then
        Sample random mini-batch E from E
        Calculate equation (6) and update θ_train
    end if
    if t mod T_c == 0 then
        θ_target ← θ_train
    end if
end for
```

Our LECRA is summarized in Algorithm 1. We use $Q(s, a; \theta)$, $Q^*(s, a; \theta^*)$ and θ^* to denote the Q value, the optimal Q value, and the optimal parameters. The replay memory of an agent stores experiences by interacting with the environment, and these experiences form a memory Dataset $\mathcal{E}$, with the maximum size of B. In Algorithm 1, the Mean Squared Error (MSE) is adopted to calculate the loss, and the loss of a random mini-batch E is defined as:

$$L(\theta_{train}^{(t)}) = \frac{1}{b} \sum_{e^{(t)}\in E} (y_{target}^{(t)} - Q(s^{(t)}, a^{(t)}; \theta_{train}^{(t)}))^2 \quad (6)$$

where the target value $y_{target}^{(t)}$ for DQN training is calculated as:

$$y_{target}^{(t)} = r^{(t+1)} + \gamma \max_{a'} Q(s^{(t)}, a'; \theta_{target}^{(t)}) \quad (7)$$

At each time slot, the stochastic gradient descent algorithm is applied to train DQN by using mini-bath $\mathcal{E}$.

IV. Simulation Results

A. AAAN Setup

The AAAN consists of ten communication links ($N = 10$) that share a limited number of frequency channels ($K = 2, 3, 5$). To simulate the 3-D dynamic airspace, all A2A communication pairs are randomly located within a cube of

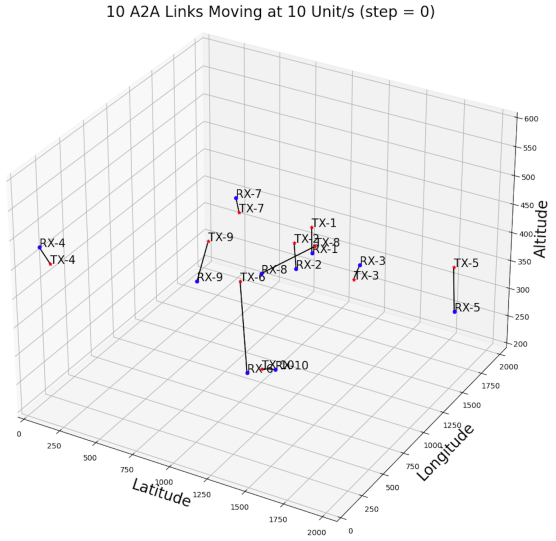


Fig. 3: 10 A2A Links Initial Deployment in a 3-D Space.

$2,000 \times 2,000$ in horizontal distance and between 300 and 500 in altitude. The distance between the transmitter and receiver of each communication link is in the range of $(20, 500)$. Fig. 3 provides a 3-D view of the initial random deployment of the ten A2A links. In our simulation, each time slot corresponds to the channel coherence time that is set as 20 ms.

B. Model Parameters

LECRA provides two different neural network policies: DQN and DDPG+DQN. For both polices, we set the maximum number of neighbors as $N_{neighbor} = 5$. In Policy 1, the number of transmit power levels is set as $\delta = 8$, and there are three hidden layers of size (200, 200, 100). Policy 2 consists of three different networks: Critic, Actor, and DQN networks. Both the Critic and Actor networks have three hidden layers of size (200, 150, 100), while the DQN network also has three hidden layers of size (150, 150, 100).

In our experiments, each hidden layer takes ReLU as an activation function. We apply RMSProp as the optimizer to minimize the loss calculated by equation (6), where adaptive learning rate is applied. The DRL model applies adaptive ϵ-greedy strategy with an initial value $\epsilon_0 = 0.05$, which is suppressed by the decay factor λ_ϵ. The replay memory size is $B = 1000$, and the mini-batch size is $b = 128$. To maintain "quasi-static", the hyper-parameter of the target network is updated every 50 time steps ($T_c = 50$). The discount factor is set as $\gamma = 0.25$. For the reward function in (3), we assign $\lambda_{SINR}^{(t)} = 8$ to punish each link violating the minimum SINR requirement.

C. Performance Evaluation

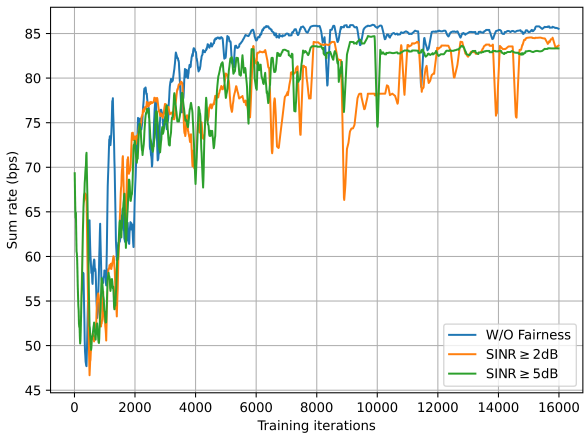
We conduct extensive experiments to evaluate the performance of our proposed LECRA algorithm. In particular, we set the moving speed of aircraft at $v = 10$ $units/s$ and the moving direction is randomly selected at the beginning of each experiment.

a) Sum-rate Vs. User Fairness: In this set of experiments, we evaluate the sum-rate of the AAAN under individual QoS constraints. We assume ten A2A links share three frequency channels. Fig. 4a and 4b depict the learning results for Policy 1 and 2 respectively. In each plot, we compare the LECRA performance for three cases: no fairness constraint, $SINR \geq 2dB$ and $SINR \geq 5dB$. Our observations can be summarized as follows: (1) Policy 1 on average achieves higher sum rates than Policy 2. Specifically, in Fig. 4a the sum-rates of Policy 1 in the three cases converge to 85.7 bps, 83.2 bps, and 83.1 bps respectively. In Fig. 4b the sum-rates of Policy 2 converge to 85.9 bps, 84.0 bps, and 83.3 bps respectively. (2) Policy 2 converges faster than policy 1. We see that Policy 1 starts to converge after about $5,000$ training iterations, while Policy 2 can converge after about $3,000$ training iterations. For both policies, due to the high mobility of the AAAN, all learning curves experience fluctuations after converging. (3) Generally, as the QoS threshold $SINR^{min}$ increases, the sum rate decreases and becomes more fluctuated.

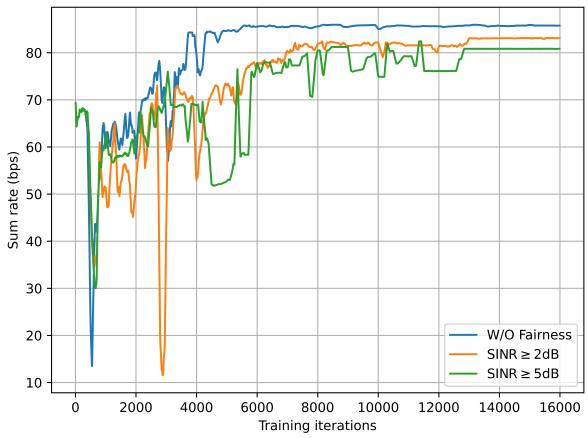
b) Sum-rate Vs. Number of Channels: In this set of experiments, we investigated the relationship between the sum rate and the number of available channels. For easy exposition, we assume ten A2A links share either two or five channels without user fairness constraint. Fig. 5 depicts the learning results for both policies. As expected, both the sum rate and convergence time increase with the number of available channels, due to added spectrum resource and searching space. Specifically, for $K = 2$, Policy 1 and 2 converge to the highest sum rate at 74.1 bps and 74.3 bps after about 3000 training iterations. For $K = 5$, Policy 1 and 2 converge to the highest sum rate at 99.7 bps after about 6000 training iterations. We also observe that Policy 1 introduces more fluctuations than Policy 2. This is because the DQN structure in Policy 1 has finite discrete action space for both channel and power allocation. In particular, we discretize the transmission power range into eight levels for the power control action space. However, it is likely that the optimal power control is not included in this discretized action space. As a result, Policy 1 may switch back and force to find out the optimal power value. On the contrary, DDPG has continuous action space to find out the optimal power control strategy.

V. Conclusions

In this paper, the problem of dynamic resource allocation was studied to maximize the aggregate spectrum utilization efficiency in a multi-channel AAAN. We proposed a DRL-based edge computing and communication solution to autonomously find the optimal channel selection and power control strategy. Our LECRA algorithm allows the agent of each A2A communication link to learn the optimal policy in a distributed manner by gathering and analyzing only the local information. The experimental results demonstrated the effectiveness of our LECRA algorithm in complex and dynamic AAAN environments.



(a) Policy1: DQN.



(b) Policy2: DDPG+DQN.

Fig. 4: 10 Links (Speed = 10 $units/s$) Share 3 Channels with/without SINR Fairness.

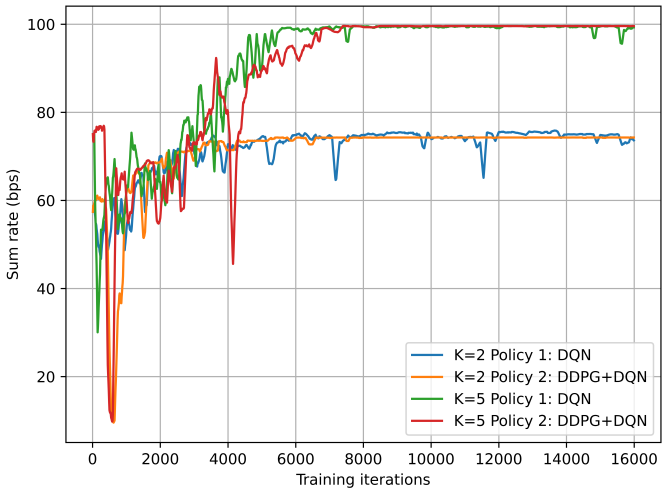


Fig. 5: 10 Links (Speed = 10 $units/s$) Share Various Channels without SINR Fairness.

REFERENCES

[1] D. L. Hackenberg, "UAM coordination and assessment team (UCAT) NASA UAM update for ARTR," in *National Academies UAM Study Kickoff for the Aeronautics Research and Technology Roundtable*, 2019.

[2] R. D. Apaza, E. J. Knoblock, and H. Li, "A new spectrum management concept for future nas communications," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, pp. 1–7, IEEE, 2020.

[3] E. J. Knoblock, R. D. Apaza, H. Li, Z. Wang, R. Han, N. Schimpf, and N. P. Rose, "Investigation and evaluation of advanced spectrum management concepts for aeronautical communications," in *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, pp. 1–12, 2021.

[4] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[5] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Transactions on Wireless Communications*, vol. 18, no. 1, pp. 310–323, 2018.

[6] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[7] L. Liang, J. Kim, S. C. Jha, K. Sivanesan, and G. Y. Li, "Spectrum and power allocation for vehicular communications with delayed csi feedback," *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 458–461, 2017.

[8] Z. Zhou, J. Feng, L. Tan, Y. He, and J. Gong, "An air-ground integration approach for mobile edge computing in IoT," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 40–47, 2018.

[9] B. Shang and L. Liu, "Mobile-edge computing in the sky: Energy optimization for air–ground integrated networks," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7443–7456, 2020.

[10] W. Zhang, L. Li, N. Zhang, T. Han, and S. Wang, "Air-ground integrated mobile edge networks: A survey," *IEEE Access*, vol. 8, pp. 125998–126018, 2020.

[11] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency minimization for D2D-enabled partial computation offloading in mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4472–4486, 2020.

[12] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 3, pp. 1750–1763, 2019.

[13] A. A. Ateya, A. Muthanna, and A. Koucheryavy, "5G framework based on multi-level edge computing with D2D enabled communication," in *2018 20th International Conference on Advanced Communication Technology (ICACT)*, pp. 507–512, 2018.

[14] Z. Wang, K. E. J. Li, Hongxiang, and R. D. Apaza, "Joint spectrum access and power control in air-air communications - a deep reinforcement learning based approach," in *2021 40th Digital Avionics Systems Conference (DASC)*, 2021.

[15] R. Wang, J. Zhang, S. Song, and K. B. Letaief, "Optimal qos-aware channel assignment in D2D communications with partial csi," *IEEE Transactions on Wireless Communications*, vol. 15, no. 11, pp. 7594–7609, 2016.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.