Dynamic Spectrum Sharing in Cellular Based Urban Air Mobility via Deep Reinforcement Learning

Ruixuan Han[*], Hongxiang Li[*], Eric J. Knoblock[†], Michael R. Gasper[†], Rafael D. Apaza[†]
[*]Department of Electrical and Computer Engineering, University of Louisville, Louisville, KY, United States
[†]Communications and Intelligent Systems Division, NASA Glenn Research Center, Cleveland, OH, United States

GLOBECOM 2022 - 2022 IEEE Global Communications Conference | 978-1-6654-3540-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/GLOBECOM48099.2022.10001062

***Abstract*—With increasing mobility demands in metropolitan areas, the emerging concept of Urban Air Mobility (UAM) opens a new urban air transportation paradigm, where one big challenge is to ensure reliable two-way communications between aerial vehicles and their ground control stations for safe operations. The concept of cellular-based UAM (cUAM) provides a promising solution where aerial vehicles are regarded as new aerial users, sharing the cellular spectrum with existing terrestrial users. However, with new characteristics and demands of aerial users, the severity of spectrum scarcity becomes more prominent and new spectrum management solutions are needed. In this paper, we consider a typical cUAM scenario where multiple aerial vehicles transport passengers/cargo along their pre-defined paths, with the coexistence of terrestrial users. We assume the minimum communication Quality of Service (QoS) must be achieved for all users at all times. Our objective is to simultaneously minimize aerial users' mission completion time and maximize terrestrial users' achievable data rate by jointly optimizing the spectrum allocation for all users and the moving velocities of aerial users. We formulate the optimization problem as a multi-stage Markov Decision Process and propose a multi-agent deep reinforcement learning-based algorithm. We also propose a heuristic greedy algorithm and an orthogonal multiple access algorithm as baseline solutions. Simulation results show that our learning-based solution outperforms the baseline solutions under different network configurations.**

I. Introduction

With the rapid growth of population and urbanization, the new concept of Urban Air Mobility (UAM) has drawn significant attention to address the mobility challenges in metropolitan areas [1]. In particular, by leveraging the latest advancements in new vertical takeoff and landing (eVTOL) aircraft, air traffic management, artificial intelligence (AI), and communication technologies, UAM opens a new urban transportation paradigm [2]. As in other aviation systems, all UAM aerial vehicles (AVs) need to maintain reliable two-way communications with their ground Air Traffic Control (ATC) stations to ensure safe operations. Recently, the cellular-connected UAM (cUAM) has been envisioned as a promising solution to provide communication services for urban air transportation systems, where each AV is integrated as an aerial user into the cellular system, sharing the spectrum with existing terrestrial users (TUs). The cUAM solution provides not only licensed spectrum but also ubiquitous coverage and connectivity with widely deployed base stations (BS) [3].

On the other hand, cUAM faces new challenges. First, the cellular spectrum is already congested, so the additional aeronautical communication demands will amplify the spectrum scarcity issue. Second, since existing cellular networks are designed for terrestrial users, it's challenging to meet AVs' communication quality of service (QoS) requirements due to their high mobility and flying altitude. Therefore, new spectrum sharing solutions are needed to support future cUAM applications. To this end, the National Aeronautics and Space Administration (NASA) is leading the effort to investigate advanced spectrum management concepts to modernize the spectrum utilization in UAM related applications [4], [5].

There are existing research works on AV-related spectrum management. For example, the authors in [6] studied a joint resource allocation and trajectory optimization problem for a multi-UAV enabled downlink cellular network to maximize the throughput. Paper [7] investigated a multi-dimensional resources management problem in a UAV assisted vehicular network to maximize the number of offloaded tasks. Even though most existing works focus on data-driven applications, spectrum management in cUAM has to consider the needs of different types of users. While TUs aim to maximize their data transmission rates, the primary goal of AVs is to safely and quickly transport passengers/cargo. Therefore, we consider a hybrid optimization objective that combines TUs' achievable sum rate and AVs' total mission completion time. Some existing works on mission time minimization can be found in the literature [8]–[10], which didn't consider spectrum management. In cUAM, by minimizing AVs' mission completion time, more transportation missions can be accomplished per unit time and bandwidth, which in turn improves the spectrum utilization efficiency.

In this paper, we consider a typical cUAM scenario where multiple AVs transport passengers/cargo along their pre-defined paths, with the coexistence of TUs. During the flight, each AV needs to maintain reliable bidirectional control and non-payload communications (CNPC) with cellular base stations (BS) to ensure safe operations. Meanwhile, each TU's data transmission rate also needs to be satisfied. We aim to maximize the revised spectrum utilization efficiency, defined as TUs' weighted sum rate minus AVs' total travel time, by jointly optimizing AVs' velocities and all users' channel allocation. Essentially, it is a multi-stage non-convex combinatorial problem, and finding the optimal solution is quite challenging. To solve it, we transform the joint optimization problem into a Markov Decision Process (MDP) and propose a multi-agent deep reinforcement learning (MADRL) algorithm. Simulation results show that MADRL outperforms other non-

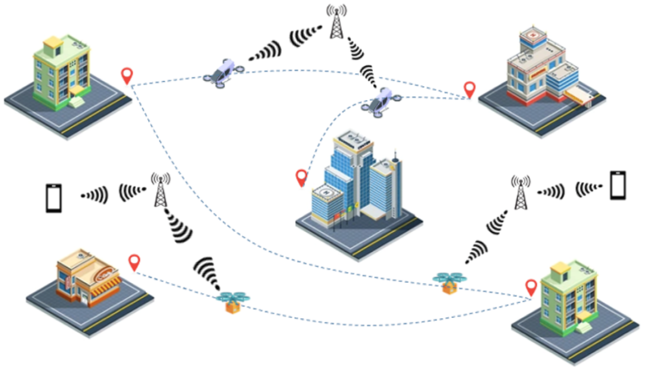


Fig. 1. The cUAM application.

learning based benchmarks (i.e., orthogonal multiple access algorithm and heuristic greedy algorithm).

II. System Model and Problem Formulation

A. System Model

We consider a cUAM scenario shown in Fig. 1, where a set of AVs $\mathcal{N}_a = \{1,2,...,N_a\}$ transport passengers/cargo along their pre-defined paths, with the coexistence of stationary TUs $\mathcal{N}_t = \{1,2,...,N_t\}$. The entire user set is denoted as $\mathcal{N} = \mathcal{N}_a \cup \mathcal{N}_t$, which is served by a set of BSs $\mathcal{B} = \{1,2,...,B\}$ with a set of frequency channels $\mathcal{K} = \{1,2,...,K\}$. We assume all AVs fly at the same altitude so the location of AV n at time t can be denoted by its horizontal coordinates $\mathbf{q_n(t)} = [x_n(t), y_n(t)]^T$. The total travel time is discretized into time slots $\mathcal{T} = \{1,2,...,T\}$ of duration Δ_t. We assume all AVs are synchronized and each time slot is split for uplink and downlink transmissions in time division duplexing (TDD) mode. Due to spectrum scarcity, we assume the same frequency channel can be shared by multiple users as long as the co-channel interference level is controlled under an acceptable level.

In uplink transmissions (from users to BSs), each BS may receive signals from both the desired and undesired users, causing co-channel interference. For user n at time t, we let $a^{up}_{n,k}(t)$ be the binary channel allocation indicator (i.e., $a^{up}_{n,k}(t) = 1$ when channel k is allocated to user n; otherwise $a^{up}_{n,k}(t) = 0$), $\mathcal{U}^{up}_n(t)$ be the set of users sharing the same uplink channel with user n, and $h_{n,b}(t)$ be the channel power gain between user n and BS b. The received signal-to-interference-plus-noise power ratio (SINR) at user n's associated BS is given by:

$$\gamma^{up}_n(t) = \frac{p h_{n,b_n(t)}(t)}{\sum_{u \in \mathcal{U}^{up}_n(t)} p h_{u,b_n(t)} + N_0} \tag{1}$$

where p is AV's transmit power, $b_n(t)$ is the index of user n's associated BS, and N_0 denotes the additive white Gaussian noise (AWGN) power spectral density. Let W denote the channel bandwidth, user n's uplink data rate at time t can be calculated as:

$$r^{up}_n(t) = W \log_2(1 + \gamma^{up}_n(t)) \tag{2}$$

Similarly, in downlink transmissions, users receive signals from both associated and non-associated BSs, causing co-channel interference. The downlink data rate for user n at time t can be calculated as:

$$r^{dw}_n(t) = W \log_2(1 + \gamma^{dw}_n(t)) \tag{3}$$

where $\gamma^{dw}_n(t)$ is user n's received downlink SINR at time t. Considering spectrum scarcity and user fairness, we assume each user can use only one channel at any time. Accordingly, we have

$$\sum_{k \in \mathcal{K}} a^{up}_{n,k}(t) = 1, \sum_{k \in \mathcal{K}} a^{dw}_{n,k}(t) = 1, \forall n \in \mathcal{N} \tag{4}$$

B. Problem Formulation

Since the goal of cUAM is to jointly minimize AVs' mission completion time and maximize TUs' sum rate, we design the utility function as TUs' weighted sum rate minus AVs' total travel time and aim to maximize it by jointly optimizing the spectrum allocation $\mathbf{A} = \{a_{n,k}(t), n \in \mathcal{N}, k \in \mathcal{K}\}$ and AVs' trajectories $\{\mathbf{q}_n(t), n \in \mathcal{N}_a\}$. With pre-defined AV paths, the trajectories are solely determined by AVs' velocities. Let T_n denote AV n's travel time and $\mathbf{V} = \{v_n(t), n \in \mathcal{N}_a\}$ denote AVs' moving velocities, the optimization problem can be formulated as:

$$\begin{aligned}(P1): \max_{\mathbf{V},\mathbf{A}} \quad & -\sum\nolimits_{n \in \mathcal{N}_a} T_n + \alpha \sum\nolimits_{m \in \mathcal{N}_t} (r^{up}_m(t) + r^{dw}_m(t)) \\ \text{s.t.} \quad & C1: \gamma^{up}_n(t) \geq \gamma^{up}_{qos}, \gamma^{dw}_n \geq \gamma^{dw}_{qos}, \forall n \in \mathcal{N} \\ & C2: \sum_{k \in \mathcal{K}} a^{up}_{n,k}(t) = 1, \sum_{k \in \mathcal{K}} a^{dw}_{n,k}(t) = 1, \forall n \in \mathcal{N} \\ & C3: \mathbf{q}_n(0) = \mathbf{q}^I_n, \mathbf{q}_n(T_n) = \mathbf{q}^F_n, \forall n \in \mathcal{N}_a \\ & C4: \|\mathbf{q}_n(t) - \mathbf{q}_n(t-1)\|_2 \leq V_{max}, \forall n \in \mathcal{N}_a \\ & C5: \|\mathbf{q}_n(t) - \mathbf{q}_{n'}(t)\|_2 \geq d_{min}, \forall n, n' \in \mathcal{N}_a\end{aligned}$$

In $(P1)$, constraint $C1$ guarantees the minimum SINR requirements for uplink and downlink communications. Constraint $C2$ requires that each AV chooses one and only one channel at any time. Constraint $C3$ enforces that each AV moves from its source to destination, where $\mathbf{q}^I_n$ and $\mathbf{q}^F_n$ indicates AV n's initial and final locations. Constraint $C4$ specifies AV's maximum speed limit. Constraint $C5$ implements the collision avoidance. Note that solving $(P1)$ is quite challenging since it is a non-convex multi-stage combinatorial optimization problem that suffers from the curse of dimensionality.

III. MADRL-based Joint Spectrum and Velocity Optimization

In this section, we develop a MADRL algorithm to solve problem $(P1)$. Specifically, we first transform the joint spectrum and velocity optimization problem into a MDP, and then propose a cooperative MADRL solution.

A. MDP Formulation

To apply MADRL to the cUAM scenario, we first need to re-model the optimization problem $(P1)$ as a MDP. The objective of MDP is to find a good policy for the agent to maximize the accumulated reward. MDP provides a rigorous mathematical framework to model the sequential decision making process. In each step of the MDP, the agent interacts with the environment and gets more insights about the environment according to the environment's feedback. In particular, four elements are included in the agent-environment interaction: state s, action a, reward r, and state transition function $\mathcal{P}$. During the interaction, the agent first observes the environment state s and then takes an action a based on that state s. The environment receives the action and transits into the next state s' according to the state transition function $\mathcal{P}$. Meanwhile, the environment generates a reward r as feedback to inform the agent how much the objective has been achieved. In cUAM, all users are RL agents, and the environment is the entire cellular network including all user locations, BSs and available frequency channels. The MDP formulation for problem $P1$ is described as follows:

- State $\mathcal{S}$: Due to the existence of multiple users in the cUAM network, problem $P1$ can be formulated as a multi-agent MDP, where each agent has its own state observation. Let $\mathbf{s}_n$ denote the state observation by agent n, it includes (1) agent's type; (2) self mission completion status; (3) other users' location related information. Note that this is the information needed for each user's decision making. Specifically, $\mathbf{s}_n$ can be represented as:

$$\mathbf{s}_n = (e_n, o_n, \{d_j, d_{n,b_j}, \mathbf{b}_j\}_{j\in\mathcal{N}, j\neq n},) \tag{5}$$

  The first part $e_n \in \{0,1\}$ indicates the agent's type (i.e., AV or TU). The second part $o_n \in \{0,1\}$ indicates whether agent n has reached its destination or not. The third part $\{d_j, d_{n,b_j}, \mathbf{b}_j\}_{j\in\mathcal{N}, j\neq n}$ represents other agents' location-related information, where $\mathbf{b}_j \in \mathbb{R}^M$ is a one-hot vector indexing agent j's associated BS, and d_j (resp. d_{n,b_j}) is the normalized distance between agent j (resp. agent n) and BS $\mathbf{b}_j$.
- Action $\mathcal{A}$: Since TU agents are stationary, they only need to decide channel selections. However, AV agents' action space includes both velocity and channel selections. To be consistent, we have the same action space for all agents and enforce zero speed for TU agents. Accordingly, the joint action for agent n can be denoted as $\mathbf{a}_n = (a_{v,n}, a_{c,n})$, where $a_{v,n}$ and $a_{c,n}$ represent the velocity and the channel selections. Since velocity selection can be continuous but channel selection is discrete, we discretize the continuous velocity into L speed levels such that both the velocity and channel selections can be integrated into one discrete action space. Note that the channel selections for uplink and donwlink are independent. The total number of available actions is LK^2.
- State Transition Function $\mathcal{P}$: Since the state is determined by all agents' locations, we define the mapping function $f : (\mathbf{q}_n, \{\mathbf{q}_j\}_{j\in\mathcal{N}_a, j\neq n}) \mapsto \mathbf{s}_n$. After each time step, AV n's new location can be calculated as

$$\mathbf{q}'_n = \mathbf{q}_n + \Delta_t \mathbf{a_{v,n}} \tag{6}$$

  Therefore, the next state observed by AV n becomes $\mathbf{s}'_n = f(\mathbf{q}'_n, \{\mathbf{q}'_j\}_{j\in\mathcal{N}_a, j\neq n})$, and the state transition function is given by $\mathcal{P} : \mathbf{s}_n \mapsto \mathbf{s}'_n$.
- Reward Function $\mathcal{R}$: The reward is a numerical value obtained by the agent from the environment, and it quantifies how much the agent's objective has been achieved. For the cUAM system, the reward for each agent at each time step is defined as:

$$r = r_f + r_d + r_v + \alpha r_m \tag{7}$$

  In Eq. (7), reward r consists of four parts: (1) AV's mission completion reward r_f. An AV agent receives a fixed reward r_f when completing its mission (i.e., reaching the destination). (2) AV's distance based mission incompletion penalty r_d. It is a negative number proportional to AV's remaining distance to the destination. This penalty forces AVs to move towards their destinations as quickly as possible. (3) Constraint violation penalty r_v. It is imposed when an agent's action violates constraint $C1$ or $C5$ in $P1$. (4) TU's weighted achievable rate reward r_m. A higher data rate leads to a larger reward.

With the above MDP formulation, for an episode of T time steps, maximizing the utility function in $(P1)$ becomes maximizing the accumulated rewards $G = \sum_t \sum_{n\in\mathcal{N}_a} r_n(t)$. This MDP can be considered as an episodic task, which has a terminal state that separates the agent-environment interactions into episodes. In each episode, the task starts at the initial state (i.e., AVs' initial locations) and ends when all AVs reach their destinations.

B. MADRL Algorithm

Our proposed MADRL algorithm combines Value Decomposition Networks (VDN) [11] with Dueling Double Deep Q Network (D3QN) [12] to learn joint actions. Specifically, D3QN is adopted as the decision engine for each agent to generate individual Q-value. Then, VDN combines all individual Q-values and learns a joint Q-value that reflects the quality of the joint action. Therefore, as a global controller, VDN enables cooperation among all agents.

1) D3QN: D3QN is a value-based RL algorithm, where the optimal policy is achieved according to the state-action value $Q(s,a)$. Specifically, D3QN takes the state observation s as the input, and utilizes a deep neural network (DNN) parameterized by θ to generate state-action value $Q(s,a;\theta)$ as the approximation of the true state-action value $Q(s,a)$. Different from the standard DQN [13], D3QN utilizes dueling architecture and double learning technique to improve Q-value approximation accuracy. With the dueling architecture, D3QN separately estimates the state-value and the advantage for each action, and then combines them via an aggregation layer to generate an estimation of the state-action value $Q(s, a)$. Such a dueling architecture can help D3QN quickly identify the appropriate action during the training process. With the double learning technique, as shown in Eq. (9), D3QN can mitigate the state-action value overestimation problem in standard DQNs. To train the DNN parameters θ, D3QN samples transitions $< s, a, r, s' >$ from the experience replay buffer $\mathcal{D}$. Each transition indicates an agent-environment interaction, where s' is the observed state after taking action a at state s and r is the reward. Parameters θ are updated by minimizing the squared TD error:

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}[(y^{D3QN} - Q(s, a; \theta))^2] \tag{8}$$

with

$$y^{D3QN} = r + \gamma Q(s', \arg\max_{a'} Q(s', a'; \theta); \theta^-) \tag{9}$$

where γ is the discounted factor, and θ^- represent those parameters of the target network. Note that the target network has the same structure as the action-value neural network, and it periodically copies values from θ to make the training procedure stable.

2) VDN: In cUAM, since coexisting AVs and TUs share the same set of frequency channels, all users are interdependent and shall cooperate to achieve the common objective of maximizing the system utility function in $(P1)$. In this case, it is problematic to simply decompose the multi-agent problem $P1$ into multiple parallel single-agent problems. For example, if a user chooses its action based on its local state-action values (i.e., without knowing other users' actions), the communication QoS or collision avoidance constraint may be violated. Therefore, a global state-action value estimator is necessary to enable cooperation among all cUAM users.

In this paper, VDN [11] is adopted to estimate the global state-action value $Q_{tot}(\mathbf{s}, \mathbf{a})$. Specifically, VDN utilizes a value-decomposition layer to represent the global state-action value as the sum of individual action-action values across all agents, and it can be represented as:

$$Q_{tot}(\mathbf{s}, \mathbf{a}) = \sum_{n=1}^{N} Q_n(s_n, a_n; \theta) \tag{10}$$

where s_n, a_n and Q_n are agent n's individual state observation, action and state-action value; $\mathbf{s} = (s_1, s_2, ..., s_N)$ and $\mathbf{a} = (a_1, a_2, ..., a_N)$ represent the joint state observation and action. During the training process, the DNN parameters in each agent are updated by the global Q-value rather than its local Q-value. In this way, the value-decomposition layer enables cooperation among agents such that better joint actions can be made to improve the spectrum utilization efficiency.

Algorithm 1: MADRL-based joint optimization for spectrum allocation and velocity selection

```
1  Initialize $\theta$, $\epsilon = \epsilon_0$, and $\theta^- = \theta$;
2  for $n_{epi} = 1, 2, ..., \hat{N}_{epi}$ do
3      Set the time step $t = 0$ and reset location $\mathbf{q}_n(t) = \mathbf{q}_n^I$ for each AV $n$;
4      while $\mathbf{q}_n(t) \neq \mathbf{q}_n^F, \forall n \in \mathcal{N}$ and $t \leq \hat{N}_{step}$ do
5          for agent $n = 1, 2, ..., N$ do
6              Obtain state $s_n$ according to mapping function $f$;
7              Choose action $a_n$ from action space $\mathcal{A}$ based on $\epsilon$-greedy policy, where $a_n = \begin{cases} \text{random action}, & \text{with prob } \epsilon \\ \arg\max_{a \in \mathcal{A}} \tilde{Q}(a, s_n; \theta), & \text{otherwise} \end{cases}$;
8          end
9          Execute action $\mathbf{a}$ and observe next state $\mathbf{s}'$;
10         Calculating total reward $r_{tot} = \sum_n r_n$;
11         Store transition $(\mathbf{s}, \mathbf{a}, r_{tot}, \mathbf{s}')$ in $\mathcal{D}$;
12         Update time step $t := t + 1$ and $\epsilon := \sigma\epsilon$;
13     end
14     Sample minibatch of $\hat{N}_b$ episodes from $\mathcal{D}$;
15     Calculate the loss function $\mathcal{L}(\theta)$;
16     Update $\theta$ by using gradient descent optimizer: $\theta := \theta - \alpha\nabla_\theta(y_{tot} - Q_{tot}(\mathbf{s}, \mathbf{a}; \theta))$;
17     After every $M$ episodes, update parameter $\theta^-$ in target network via soft update, where $\theta^- = (1 - \beta)\theta^- + \beta\theta$;
18 end
```

By combining VDN and D3QN, the proposed MADRL solution is summarized in Algorithm 1. During the training, we separate the agent-environment interactions into episodes. The procedures in each episode are shown from line 3 to line 17. At the beginning of each episode, all user locations are initialized ($\mathbf{q}_n(0) = \mathbf{q}_n^I, \forall n \in \mathcal{N}$), as shown in line 3. When all agents reach their destinations ($\mathbf{q}_n(t) = \mathbf{q}_n^F, \forall n \in \mathcal{N}$), the episode ends. Note that agents may not reach their destinations during the training process, so we impose a maximum number of steps (time slots) per episode, $\hat{N}_{step}$, as shown in line 4. In each step of the episode, the agent utilizes $\epsilon-$greedy algorithm to choose its own action a_n either randomly with a small probability ϵ or based on its state-action value $Q_n(a_n, s_n)$, as shown in line 7. According to the current state $\mathbf{s}$ and action $\mathbf{a}$, the environment transits to the next state $\mathbf{s}'$, as shown in line 9. Since all agents have a common objective, the joint total reward r_{tot} is calculated in line 10 as the sum of all agents' individual rewards: $r_{tot} = \sum_n r_n$, where r_n is defined in Eq. (7). After that the transition $< \mathbf{s}, \mathbf{a}, r_{tot}, \mathbf{s}' >$ is stored into the reply buffer $\mathcal{D}$ (line 11). Then, the time step and ϵ are updated in line 12. At the end of each episode (line 14 to 17), the DNN in each agent $Q_n(s, a; \theta)$ is trained by the stored transitions. In particular, parameter θ is updated by minimizing the following loss function:

$$\mathcal{L}(\theta) = \frac{1}{\hat{N}_b} \sum_{i=1}^{\hat{N}_b} \left[(y_i^{tot} - Q_{tot}(\mathbf{s}, \mathbf{a}; \theta))^2\right] \tag{11}$$

with

$$y^{tot} = r_{tot} + \gamma Q(\mathbf{s}', \arg\max_{\mathbf{a}'} Q_{tot}(\mathbf{s}', \mathbf{a}'; \theta); \theta^-) \tag{12}$$

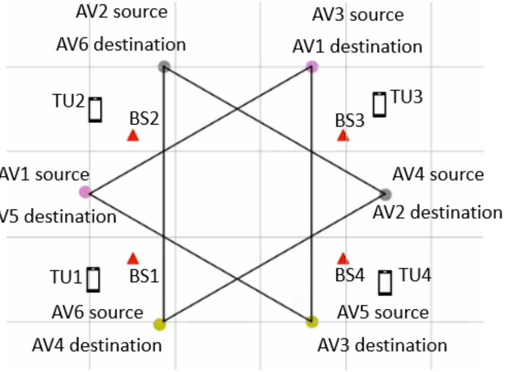


Fig. 2. Simulated cUAM scenario.

where $\hat{N}_b$ is the number of episodes sampled from the replay buffer, and θ^- are parameters in the target network. After every M episodes, θ^- are soft updated to stabilize the training, as shown in line 17. The training process is completed when the accumulated reward converges to a stable value. Then, the trained model can be deployed in each agent to perform real-time channel allocation and velocity selection.

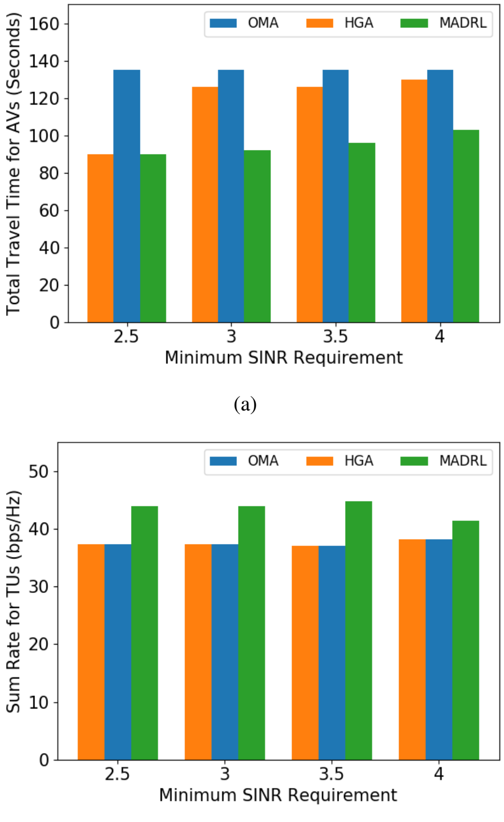
TABLE I
SIMULATION PARAMETERS.

Simulation Parameters	Value
Departure X-coordination	[0, 433, 1299, 1732, 1299, 433]
Destination X-coordination	[1299, 1732, 1299, 433, 0, 433]
Departure Y-coordination	[750, 1500, 1500, 750, 0, 0]
Destination Y-coordination	[1500, 750, 0, 0, 750, 1500]
BS X-coordination	[250, 250, 1482, 1482]
BS Y-coordination	[375, 1100, 1100, 375]
Maximum number of steps $\hat{N}_{step}$	50
Initial exploration probability	1
Reaching destination reward r_f	200
QoS violation reward r_q	-10
Batch Size $\hat{N}_b$	64
Soft update rate β	0.01
Learning rate α	0.0005
Discount factor γ	0.99
AV, BS transmit power p, p'	30dBm, 40dBm
Additive noise N_0	-96dBm
Time slot duration Δ_t	1s

IV. SIMULATION RESULTS

In this section, we provide numerical results to validate the effectiveness of the MADRL algorithm for dynamic spectrum sharing in cUAM. The simulation use case is shown in Fig. 2, which comprises six AVs (solid circles), four BSs (red solid triangles), four TUs, and four frequency channels. All AVs travel along their pre-defined paths (i.e., straight lines) from their sources to destinations. The other major simulation parameters are summarized in Table I. For AVs, we consider two speed levels (i.e., hover with zero speed or move forward with the maximum speed of $V_{max} = 100$ units/s). To demonstrate the effectiveness of our learning based approach, we also consider two non-learning-based solutions as benchmarks:

- Orthogonal Multiple Access (OMA): According to the weight α, OMA splits all channels into two sets: one for AVs and one for TUs. For TUs, channels are allocated based on their locations for maximum data rate. For AVs, each channel can only be used by one AV at any time, so AVs take turns to use orthogonal channels to move forward at full speed.
- Heuristic greedy algorithm (HGA): HGA has the same rule for TUs' channel allocation. For AVs, HGA assumes that all AVs are greedy to move forward and no more than two AVs can share the same frequency channel. Under these assumptions, HGA reformulates the original problem $P1$ into an AV paring problem, which is a binary integer programming problem and can be solved by using the PULP library in Python [14].



(a)

(b)

Fig. 3. Experimental results: (a) Mission completion time comparison, (b) Achievable rate comparison

Fig. 3 shows the performance comparison of different algorithms when $\alpha = 0.001$. Regarding AVs' total travel time in Fig. 3(a), we observe that: (1) MADRL has the least travel time under different SINR requirements. For example, when γ_{qos}=4, the total travel time for MADRL, HGA and OMA are 103, 130 and 135 seconds, where MADRL outperforms HGA and OMA by 20.77% and 23.70% respectively. (2)

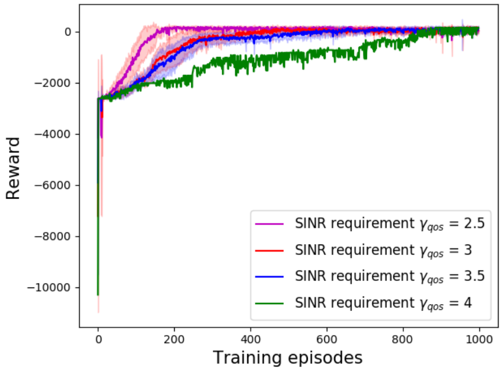


Fig. 4. Accumulated reward per episode.

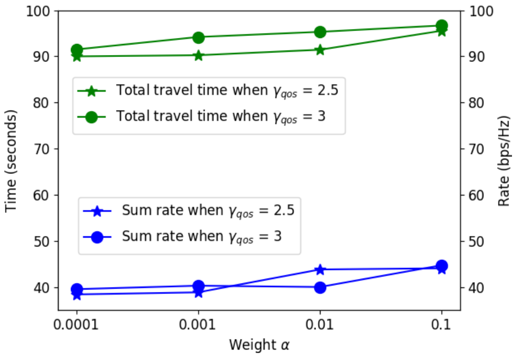


Fig. 5. The impact of the weight α

For MADRL and HGA, the travel time increases with the minimum SINR requirement. This is because, with a higher SINR requirement, AVs have to hover at certain locations for a longer time to meet the QoS constraint. (3) OMA has the worst performance with fixed travel time because it completely eliminates co-channel sharing. Regarding TUs' sum rate in Fig. 3(b), we have similar observations: (1) MADRL achieves the highest data rate in all cases. (2) OMA and HGA have the same performance because they have the same channel allocation for TUs.

Fig. 4 shows the learning curve of MADRL. We can see that MADRL converges in all cases. At the beginning of the training procedure, the DNN parameters are randomly initialized where randoms actions cause safety constraint violations leading to a large negative reward. As MADRL gradually learns from previous experiences, the accumulated reward increases with the number of training episodes.

Finally, Fig. 5 shows the impact of the weight α in MADRL. As expected, both the sum rate and the travel time increase with α. This is because, as α increases, the utility function favors more toward TUs' sum rate and less toward AVs' travel time, which leads to higher sum rate for TUs and longer travel time for AVs. Similar to Fig. 3, we can see that while the travel time increases with the minimum QoS requirement, the sum rate is less sensitive to the minimum QoS requirement.

V. Conclusion

In this paper, we studied dynamic spectrum sharing in cUAM where multiple AVs transport passengers/cargo along with their pre-defined paths, with the coexistence of multiple TUs. Due to the different needs of aerial and terrestrial users, a MDP optimization problem was formulated to simultaneously minimize AVs' mission completion time and maximize TUs' sum rate. Considering the non-convex and combinatorial nature of the problem, we proposed a MADRL algorithm that combines VDN and D3QN to jointly optimize the channel allocation and AVs' moving velocities. We also proposed two non-learning based solutions as baselines for performance comparison. Extensive simulation results showed that the MADRL-based solution outperforms the non-learning based solutions in all cases.

References

[1] B. P. Hill, D. DeCarme, M. Metcalfe, C. Griffin, S. Wiggins, C. Metts, B. Bastedo, M. D. Patterson, and N. L. Mendonca, "Uam vision concept of operations (conops) uam maturity level (uml) 4," NASA, Tech. Rep., 2020.

[2] J. Doo, M. Pavel, A. Didey, C. Hange, N. Diller, M. Tsairides, M. Smith, E. Bennet, M. Bromfield, and J. Mooberry, "Nasa electric vertical takeoff and landing (evtol) aircraft technology for public services–a white paper: Nasa transformative vertical flight working group 4 (tvf4)," NASA, Tech. Rep., 2021.

[3] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected uav: Potential, challenges, and promising technologies," *IEEE Wireless Communications*, vol. 26, no. 1, pp. 120–127, 2018.

[4] 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)*. IEEE, 2021, pp. 1–12.

[5] 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)*. IEEE, 2020, pp. 1–7.

[6] S. Yin and F. R. Yu, "Resource allocation and trajectory design in uav-aided cellular networks based on multi-agent reinforcement learning," *IEEE Internet of Things Journal*, 2021.

[7] H. Peng and X. Shen, "Multi-agent reinforcement learning based resource management in mec-and uav-assisted vehicular networks," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 131–141, 2020.

[8] C. Zhan and Y. Zeng, "Completion time minimization for multi-uav-enabled data collection," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4859–4872, 2019.

[9] S. Zhang, Y. Zeng, and R. Zhang, "Cellular-enabled uav communication: A connectivity-constrained trajectory optimization perspective," *IEEE Transactions on Communications*, vol. 67, no. 3, pp. 2580–2604, 2018.

[10] X. Mu, Y. Liu, L. Guo, and J. Lin, "Non-orthogonal multiple access for air-to-ground communication," *IEEE Transactions on Communications*, vol. 68, no. 5, pp. 2934–2949, 2020.

[11] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls *et al.*, "Value-decomposition networks for cooperative multi-agent learning," *arXiv preprint arXiv:1706.05296*, 2017.

[12] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1995–2003.

[13] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[14] S. Mitchell, M. OSullivan, and I. Dunning, "Pulp: a linear programming toolkit for python," *The University of Auckland, Auckland, New Zealand*, p. 65, 2011.