A Solution for Dynamic Spectrum Management in Mission-Critical UAV Networks

Alireza Shamsoshoara*, Mehrdad Khaledi †, Fatemeh Afghah *, Abolfazl Razi *,
Jonathan Ashdown‡, Kurt Turck ‡ *School of Informatics, Computing and Cyber Systems,
Northern Arizona University, Flagstaff, AZ, USA
Email: {alireza_shamsoshoara, fatemeh.afghah, abolfazl.razi}@nau.edu
† Mathematics & Computer Science Department, Suffolk University, Boston, MA, USA
Email: mkhaledi@suffolk.edu
‡ Air Force Research Laboratory, Rome, NY, USA
Email: {jonathan.ashdown, kurt.turck}@us.af.mil

Abstract

In this paper, we study the problem of spectrum scarcity in a network of unmanned aerial vehicles (UAVs) during mission-critical applications such as disaster monitoring and public safety missions, where the pre-allocated spectrum is not sufficient to offer a high data transmission rate for real-time video-streaming. In such scenarios, the UAV network can lease part of the spectrum of a terrestrial licensed network in exchange for providing relaying service. In order to optimize the performance of the UAV network and prolong its lifetime, some of the UAVs will function as a relay for the primary network while the rest of the UAVs carry out their sensing tasks. Here, we propose a team reinforcement learning algorithm performed by the UAV's controller unit to determine the optimum allocation of sensing and relaying tasks among the UAVs as well as their relocation strategy at each time. We analyze the convergence of our algorithm and present simulation results to evaluate the system throughput in different scenarios.

Index Terms

multi-agent systems, reinforcement learning, spectrum sharing, task allocation, UAV networks

I. Introduction

Unmanned Aerial Vehicles (UAV) have several advantages such as flexibility, agility, wide field of imaging view and low deployment cost over ground-based infrastructures; hence, are widely used in many applications such as surveillance, disaster relief, wildlife monitoring, and military applications [1]–[6]. Also, UAVs face many security challenges based on the vast area of applications. Many methods such as key sharing, authentication, authorization, and so on have been proposed for these devices in the past few years [7], [8]. In some of these missions such as disaster monitoring and military applications, the pre-allocated spectrum for UAVs is not sufficient to collect and transmit high-throughput imagery data in certain locations or times, hence an on-demand access to additional spectrum is required. In this paper, we propose a solution for securing extra spectrum for the UAV networks.

In this solution, the UAV network leases a portion of time access to the licensed spectrum of a terrestrial network. In order to compensate the spectrum access grant, some of the UAVs provide relaying service for the terrestrial network, while the rest of UAVs perform their regular sensing and transmission tasks using the leased spectrum. The allocation of relaying and sensing tasks among the UAVs as well as

Distribution A: Approved for Public Release, distribution unlimited. Case Number 88ABW-2019-2375 on 15 May. 2019. The work of F. Afghah, J. Ashdown and K. Turck was supported by AFRL.

selecting their motion strategy is performed by a high-capability drone which serves as controller unit of the entire network. This optimization is performed considering various factors such as the position of the UAVs and ground0based nodes, the quality of communication channels as well as the expected lifetime of the drones.

Cooperative spectrum leasing as a property-right model spectrum sharing solution has been studied previously in the context of cognitive radio networks [9]–[14]. The main objective of these works is maximizing the benefits of the primary network through optimal time allocation between the primary and secondary networks. The ignorance of the performance of the secondary network, can cause significant operation problems in critical missions.

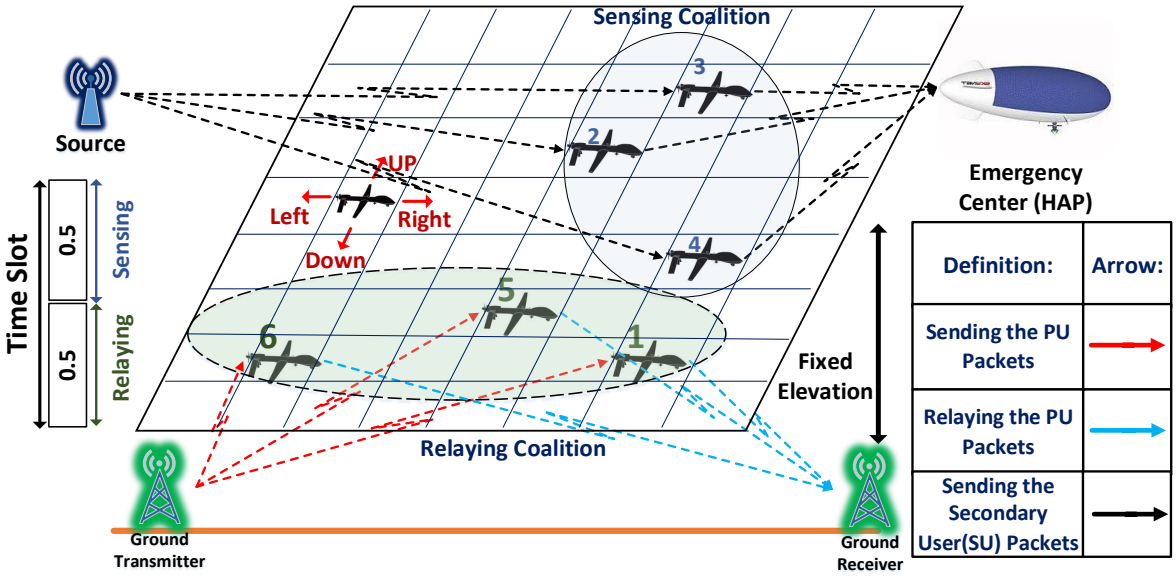

Fig. 1. A sample scheme of the proposed spectrum sharing between a UAV network and a terrestrial licensed network.

The task allocation problem to select UAVs for relaying or sensing tasks can be modeled as a multi-agent reinforcement learning (MARL) problem [15]. In a cooperative MARL problem, multiple learning agents work together to achieve a system-wide optimal solution. There have been efforts to solve task coordination in a fully distributed manner. However, the proposed solutions impose a heavy computation and decision making loads to the UAVs, and still may not converge to an optimal solution. Hence, distributed task allocation solutions are not practically feasible in highly dynamic systems [16], [17]. In this paper, we consider a remote sensing situation where the UAVs send their individual information to a UAV node designated for data fusion and overall management of the network. We assume that the UAV network does not own spectrum and opportunistically uses an external spectrum to complete the remote sensing mission. We present an algorithm based on team reinforcement learning to maximize the total utility of the system by re-positioning and determining the proper task for each drone, while minimizing the number of steps to achieve the optimal configuration. We provide experimental results to verify the convergence of the proposed algorithm.

II. System Model

Let us assume a network of N UAVs in a proximity of a licensed transmitter-receiver pair that are willing to share their spectrum with the UAV network in exchange for cooperative relaying service. The primary network is considered as a terrestrial network; while the secondary network consists of the UAVs that hover in a constant altitude. The primary pair may be in need of cooperative relaying services for experiencing a low quality direct transmission due to several reasons such as long distance between the transmitter and receiver, shadowing and fading effects. In this paper, we assume that there is no direct transmission link between the primary user (PU)'s transmitter and receiver, however, the proposal spectrum sharing method can be used in scenarios where the PU participates in spectrum leasing to take advantage of cooperative diversity gain. We assume that the UAVs are managed by a high altitude platform (HAP) UAV called as *emergency center* located in a fixed location [3].

The UAV network owns a dedicated frequency bandwidth for control commands and exchanging critical information, however it requires additional spectrum to transmit high resolution video and images during a disaster monitoring mission. In order to provide additional on-demand spectrum, the UAVs can participate in a cooperative spectrum leasing transaction with the primary network. In this case, the PU will lease half of its time access to its spectrum to the sensing UAVs, and the relay UAVs will assist the PU with forwarding their packets during the other half of the time slot. In other words, the UAVs can serve in two *sensing* and *relaying* modes. Here, we consider a centralized control scenario in which the emergency center determines the action of each UAV in terms of their motion and their role (relaying or sensing) to maximize the network utility.

Let us consider a case where K UAVs perform as relay, and the remaining $N-K$ perform the sensing task. An example of this scenario with $N = 6$ and $K = 3$ is depicted in Fig. 1. In this model, we consider a 2-D random walk for the UAVs, so that they can change their locations into adjacent cells in the grid using one of the four movement actions $\{\text{Up}, \text{Down}, \text{Left}, \text{Right}\}$. The UAVs are not allowed to take diagonal steps.

The channels between all nodes including the UAVs, the emergency center, and the PU are calculated based on pairwise distances during each time slot. Following similar works including [18], we assume that channel state information(CSI) parameters are known for the HAP UAV. Channel coefficient for each pair of users is defined as follows: i) h_{PT,U_i} is the channel parameter between the primary transmitter and i^{th} UAV, ii) $h_{U_i,PR}$ carries the channel information between the i^{th} UAV and the primary receiver, and iii) h_{S,U_i} and $h_{U_i,E}$ denote the channel parameters between the i^{th} UAV and the source and the emergency center respectively. Moreover, we assume that the noise terms at the receiver sides are normally distributed symmetric complex values, $Z \sim CN(0, \sigma^2)$.

In the literature, many papers including [19], [20] address power optimization; while we assume that the transmission power is arbitrary but constant for all transmitters.

Based on Fig. 2, we divide each time slot into two parts. We assume that, relaying UAVs receive the primary packet and forward them to the primary receiver in the first half. And in the second half, sensing UAVs forward the collected data to the emergency center. In each half slot, UAVs utilize full-duplex for the transmission.

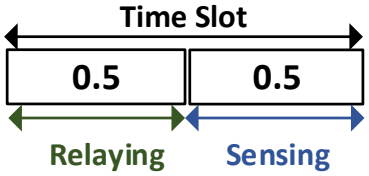

Fig. 2. One time-slot for both networks

The received signal for the UAV can be written as follow:

$$y_u[n] = h_{i,u}x_i[n] + z_r[n] \tag{1}$$

In (1), subscript i can be source or PU transmitter, x_i is the signal sent by the PU transmitter or the source and y_u is the received signal at UAV u. On the other side of the transmission, the received signal for the emergency center or the primary receiver is written as follow:

$$y_j[n] = h_{u,j}x_u[n] + z_j[n], \tag{2}$$

where subscript j can represent the emergency center or the PU-receiver, x_u is the transmitted signal by UAV u, and y_j is the received signal at the j^{th} receiver which can be the emergency center or the PU-receiver. In both (1) and (2), h_{ij} defines the effect of path loss between nodes i and j.

In this work, we utilized the Amplify and Forward (AF) relaying mode at the relay UAVs. Based on [21], [22], the throughput for the emergency center receiver can be obtained as:

$$R_{SE} = \frac{1}{2}\log_2(1 + P_S|h_{S,E}|^2 + \frac{P_S|h_{S,U_i}|^2\, P_{U_i}|h_{U_i,E}|^2}{\sigma^2 + P_S|h_{S,U_i}|^2 + P_{U_i}|h_{U_i,E}|^2}), \tag{3}$$

where P_S and P_{U_i} are transmission power for source and UAV respectively. σ^2 is the background noise power. i denotes the index for the relay(UAV). On the other hand, the throughput rate for the primary receiver is shown as:

$$R_{PU} = \frac{1}{2}\log_2(1 + P_{PT}|h_{PT,PR}|^2 + \frac{P_{PT}|h_{PT,U_i}|^2\, P_{U_i}|h_{U_i,PR}|^2}{\sigma^2 + P_{PT}|h_{PT,U_i}|^2 + P_{U_i}|h_{U_i,PR}|^2}), \tag{4}$$

where P_{PT} is the primary transmitter power. It is noteworthy that (3) and (4) are true for single relay scenarios. This system involves a multi-agent scenario including K relay UAVs and a set of $N-K$ sensing UAVs, hence the transmission rate of the relaying and sensing sub-systems can be defined as follows if $U_E = \{E_1, E_2, \dots E_{N-K}\}$ denotes the sensing set and $U_R = \{R_1, R_2, \dots, R_K\}$ is the set of relaying UAVs:

$$R_{SE}(\text{Multi-UAV}) = \frac{1}{2}\log_2(1 + P_S|h_{S,E}|^2 + \sum_{j\in U_E}\frac{P_S|h_{S,U_j}|^2\, P_{U_j}|h_{U_j,E}|^2}{\sigma^2 + P_S|h_{S,U_j}|^2 + P_{U_j}|h_{U_j,E}|^2}), \tag{5}$$

$$R_{PU}(\text{Multi-UAV}) = \frac{1}{2}\log_2(1 + P_{PT}|h_{PT,PR}|^2 + \sum_{l\in U_R}\frac{P_{PT}|h_{PT,U_l}|^2\, P_{U_l}|h_{U_l,PR}|^2}{\sigma^2 + P_{PT}|h_{PT,U_l}|^2 + P_{U_l}|h_{U_l,PR}|^2}), \tag{6}$$

where, $R_{SE}(\text{Multi-UAV})$ and $R_{PU}(\text{Multi-UAV})$ are the achievable rates using multi UAVs. Since we assumed the distances between the source and emergency center and also PT and PR are long, we can eliminate both $h_{S,E}$ and $h_{PT,PR}$ terms from the rate ($|h_{S,E}| \simeq 0$ and $|h_{PT,PR}| \simeq 0$). At the end of each time slot, the PU receiver and the emergency center announce their received throughput to the UAVs in order to provide them with a feedback (i.e., reward) based on the previous states and their taken actions.

Based on the definitions in our system model, we can define the reward or gain as a throughput difference for the whole system between to consecutive time slots such as (t) and $(t-1)$:

$$\begin{aligned} Reward = \beta_1 \times (R_{PU}(t) - R_{PU}(t-1)) \\ + \beta_2 \times (R_{SE}(t) - R_{SE}(t-1)), \end{aligned} \tag{7}$$

where, β_1 and β_2 are two control parameters. Reward in (7) is obtained based on the location of the UAVs, their tasks, and the distance to their destinations or from their sources. The goal of our proposed model is to find the best location for UAVs and the right task that result in an optimal summation throughput rate. In Section III, we use this reward function to develop a reinforcement technique for task allocation and motion planning.

III. Multi-Agent Team Reinforcement Learning

Multi-Agent Reinforcement Learning (MARL) algorithms can be categorized into three branches based on the nature of the problem. First, all the agents work cooperatively with each other. In this branch, all the agents have the same goal of maximizing the common discount factor. Also, they all have the same reward function. In the second category, the agents are fully competitive which is known as zero-sum games. In third one, the behavior of agents is a mixture of two previous groups which is neither competitive nor cooperative. Our problem in this paper is the type with fully cooperative agents. Two approaches to solve these kinds of problems are Nash Q-learning and team learning [23], [24]. We aim to use the team Q-learning for our UAVs. In the team learning approach, a single learner such as the emergency center monitors the behavior of all agents. The reason we chose this algorithm is that the single learner can utilize simple machine learning algorithms such as Q-learning in the multi-agent learning environment similar to the scenario in this paper. To model a single UAV agent in the learning environment, the Markov Decision Process (MDP) is investigated. In each step, the UAV interacts with its environment which is the state of the problem. In this specific problem, the state for the UAV is the location on the grid size. Let $S_t \in \mathcal{S}$ denote the state for each UAV at step t, where $\mathcal{S}$ is the set for all possible states. Time step t is different and it depends on the grid size of the UAVs. The i^{th} UAV at each state such as s_t^i can follow an action a_t^i based on a policy from an action set $\mathcal{A}$. The action set $\mathcal{A}$ is {Up(0), Down(1), Left(2), Right(3), Emergency Task(4), Relaying Task(5)}. The first four actions change the location for the UAV and keep the previous task from the previous location. However, the last two actions, keep the UAV in same location, but decide on its task. Action a_t^i changes S_t to $S_{t+1} \in \mathcal{S}$ for the whole system. Based on the new state S_{t+1}, the updated reward $r_t \in \mathcal{R}$ is awarded to the system based on the (7). In a single-UAV environment, the goal of the UAV is to find the optimal policy which chooses actions that maximize the $E(\sum_{t=0}^{\text{Number of Steps}} \gamma^t r_t)$ standing for expected discounted reward over time where γ is a discount factor. SARSA [25] and Q-learning [26] are two general algorithms for the RL problems. Each MPD consists of a 4-tuple $(\mathcal{S}, \mathcal{A}, P_a, R_a)$ where, $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions, $P_a : S \times a \times S' \rightarrow [0,1]$ or $P_a(S, S') = Pr(S_{t+1} = s' | S_t = s, a_t = a)$ is the probability that choosing an action a for a single UAV will change its state from S to S' at the time $t+1$. $R_a : S \times a \rightarrow \mathbb{R}$ or $R_a(S, S')$ is the expected awarded reward when a single UAV changes its state from S to S' due to action a. This immediate reward is obtained based on (7) in this special problem. Then, the single agent can update its Q-value based on:

$$\begin{aligned} Q(S,a) =& (1-\alpha) \times Q(S,a) \\ &+\alpha[R_a(S,S') + \gamma \max_{a' \in \mathcal{A}} Q(S', a')], \end{aligned} \tag{8}$$

where $\alpha (0 < \alpha \leq 1)$ is the learning rate. These equations and relations are valid for a single-learner. However, in multi-agent environment, multiple UAVs follow the taken action at the same time without

sharing any information with each other. Hence, the action vector such as $A_t = (a_1, a_2, \ldots, a_N)_{t+1} \in \mathcal{A}$ leads the system from S_t to S_{t+1} for all UAVs. Using team Q-learning, (8) can be modified and written as:

$$\begin{aligned} Q(S, [a_1, a_2, \ldots, a_N]) = (1-\alpha) \\ \times Q(S, [a_1, a_2, \ldots, a_N]) \\ +\alpha[R_a(S, S') + \gamma \max_{A' \in \mathcal{A}} Q(S', A')], \end{aligned} \tag{9}$$

where $A' = [a'_1, a'_2, \ldots, a'_N]$ is the action vector when the UAVs are placed in state S' and Q is the action-value for the whole system. There are three kinds of rewards based on the (7). If the throughput summation in a new state is greater than the previous state, then the positive reward is given to the system. If the difference is zero, then there is no reward for that (zero value). If the difference is negative between the new state and the previous state, then a negative reward is obtained by the system. The issue for team Q-learning is the process of exploration. Considering a 10×10 grid size for 2 UAVs defines 100 states for each UAVs; then, defining a team of 2 UAVs makes 10,000 states for the system just for locations without considering any tasks. This huge number of states can be time-consuming for the learning process. Action selection method is another important module for the Q-learning which stands for selecting the actions which the UAVs will perform to proceed the learning process. These methods base a fulcrum for exploration and exploitation searches [27]. In this paper, we used ϵ-greedy exploration with a constant value for the ϵ which means the system chooses random actions with the probability of ϵ and selects the best actions based on the maximum value in the Q-table with the probability of $1-\epsilon$. This method is defined in:

$$\begin{aligned} &[a_1, a_2, \ldots, a_N]_{t+1} : \\ &A_{t+1} = \begin{cases} argmax_A Q(S, A), & \text{Probability } 1-\epsilon, \\ \text{Random Actions}, & \text{Probability } \epsilon, \end{cases} \end{aligned} \tag{10}$$

where N is the number of UAVs and A is the action vector for N UAVs which brings the maximum Q value. Table I summarizes the mapping information for the team Q-learning and the definitions in this specific problem. In the past few years, deep learning has absorbed lots of attention in different areas such

TABLE I
MAPPING THE COMPONENTS BETWEEN THE TEAM Q-LEARNING AND THE SYSTEM MODEL FOR A 6×6 GRID.

Component	Description	Sample size
Actions	Making decisions about moving or the task	6
# of Agents	Number of UAVs	2
State	Considering a grid for all agents together	$(36)^2 = 1,296$
Q-Table	State(Location), Action(Movement or Task) values	$(36 \times 6)^2 = 46,656$
Reward	Difference of throughput	$R(S, A, S') \rightarrow \mathbb{R}$

as biomedical signal processing [28]–[30], pattern recognition, computer vision, autonomous vehicles, and natural language processing. Moreover, in recent studies, it has been shown that deep learning tools can be utilized alongside with reinforcement learning challenges to give a bright consciousness of the environment to the agents [26], [31]–[34].

IV. Simulation Results

In the following simulations, we consider a ground-based PU transmitter-receiver pair as well as a pair of drone-based source and emergency center with randomly initialized locations. We split the tiled coverage area into $L_1 \times L_2$ tiles, where drones are allowed to move in four directions $\{\text{Up}, \text{Down}, \text{Left}, \text{Right}\}$. Channels between nodes are calculated based on the $h_{i,j} \sim CN(0, d_{i,j}^{-2})$, where $d_{i,j}$ is Euclidean distance between nodes i and j. We arbitrarily use $\beta_1 = 1$, $\beta_2 = 1$ (defined in Eq. 7), $\alpha = 0.1$, $\gamma = 0.3$, $\epsilon = 0.1$, $\sigma^2 = 1nW$, $P_{PT} = 10mW$, $P_S = 20mW$, and P_{U_i}=$20mW$, unless otherwise specified explicitly. In the ϵ-greedy exploration, we consider a constant rate for the exploration and exploitation phases. The number of states is $(L_1 \times L_2)^N$ for N drones. We execute the MARL algorithm for 200 episodes to fill in the Q-tables for all drones based on their actions and locations. We use sequential episodes, where the resulting Q-tables for each episode is used to initialize Q-tables for the next episode. The number of steps per episode to cover the entire state space is calculated based on the experience and the number of states. We then repeat each experiment in 10 external iterations for enhanced accuracy and avoiding bias to initialization. Timing value for each scenario is provided for Python implementation with Intel Xeon(R) CPU @ 3.50GHz and 16.0GB RAM. The simulation parameters are summarized in Table II.

TABLE II
Simulation parameters and required times for the simulation with 2 UAVs and 6 actions.

Grid Size:	2×2 (4)	3×3 (9)	4×4(16)	5×5 (25)	6×6 (36)	8×8 (64)	10×10 (100)
Number of States:	4×4=16	9×9=81	16×16=256	625	1,296	4,096	10,000
Size of the Q-Table:	576	2,916	9,216	22,500	46,656	147,456	360,000
Number of Iterations:	10	10	10	10	10	10	10
Number of Episodes in each Iteration:	200	200	200	200	200	200	200
Number of Steps in each Episode:	120	800	2,000	5,000	12,000	20,000	30,000
Process Time for each epoch (Sec):	0.102628	0.762713	2.425558	8.757846	49.007077	146.936207	291.176670
Total required time:	205.256s	25.4m	80.85m	4.8h	27.2h	3 days	6 days

Figure 3 shows the sum utility for primary and emergency network and the summations of those for each episode. The sum utility represents the accumulated utility during all steps of the episode. We observe that due to using sequential learning, the latter episodes yield a higher sum utility suggesting that the algorithm converges to its final value faster. This behavior is valid for both primary and secondary networks.

Figure 4 demonstrates the number of times that UAVs switched their task and also the number of times they changed their locations in each episode. Each value at each point for a specific episode is the summation for both UAVs for all taken steps. It is logical that the HAP starts searching all states by switching tasks in each location to get a new reward. However, as simulation goes on, the HAP starts learning from its past experiences and switches the tasks for the UAV networks fewer compared to the beginning of each iteration. Also, HAP forces the UAVs to explore more locations in smaller episodes. For larger episodes, the number for movements or changing location is less than the beginning of the iteration. These behaviors demonstrate the learning process toward the optimal location and task to gain a higher throughput rate for both networks. Fig. 5 shows the required steps to get 90% of the throughput summation in each episode. The number of steps is decreasing when the number of episodes is increasing which explains the learning behavior and denotes that in less necessary steps, the emergency center can find the best states (Location) and actions for all UAVs to get the optimal reward. All figures in this section are sketched for the grid 6×6 (36). Based on Table II, the process time for each iteration is increasing exponentially for larger grid size. Moreover, this process time is dependent to the number of UAVs. This happens due to the nature of the team reinforcement learning. Since we are using distance-based deterministic channel coefficients, offline learning methods can be used to fill in the Q-tables.

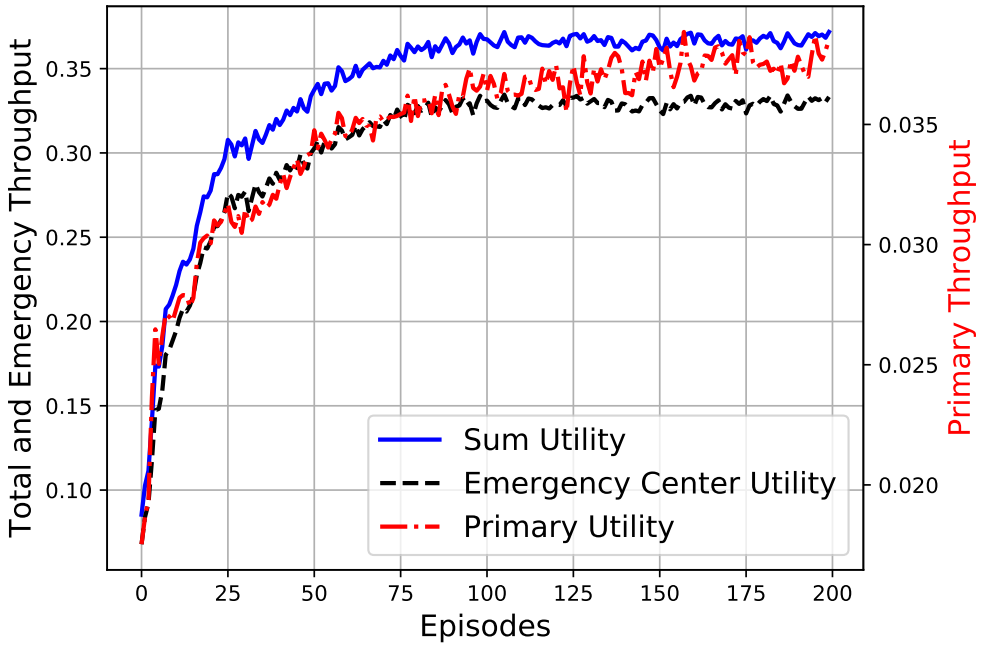

Fig. 3. Summation throughput for emergency, primary, and both networks for all UAVs.

V. Conclusion

In this paper, we introduced a solution for dynamic spectrum sharing in which a network of search-and-rescue UAVs needs an additional spectrum to send critical information such as videos to the emergency center. In this method, some of the UAVs provide a relaying service for a primary network in exchange for borrowing an additional required spectrum. We proposed a team-learning MARL algorithm, in which the emergency center of the UAV networks can determine the optimal actions of the UAVs in terms of their motion as well as the optimum task. It is worth mentioning that, in practice, the UAV network is trained by this learning mechanism in an offline manner to identify the best set of actions for different situations and use this knowledge for fast decision making during the missions. Simulation results showed the proposed distributed method converges to the optimal locations and tasks for both the primary and emergency network. The optimal state is the win-win solution for both groups. For the future works, we are working on other learning methods to reduce the effect of states (locations and tasks) and agents (UAVs) on the performance of the algorithm. Due to the exponential behavior of team learning, it takes time for offline learning; however, we aim to increase the number of UAVs and the grid size at the same time.

VI. Acknowledgment of Support and Disclaimer

The authors acknowledge the U.S. Governments support in the publication of this paper. This material is in part based upon work funded by AFRL. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the US government or AFRL.

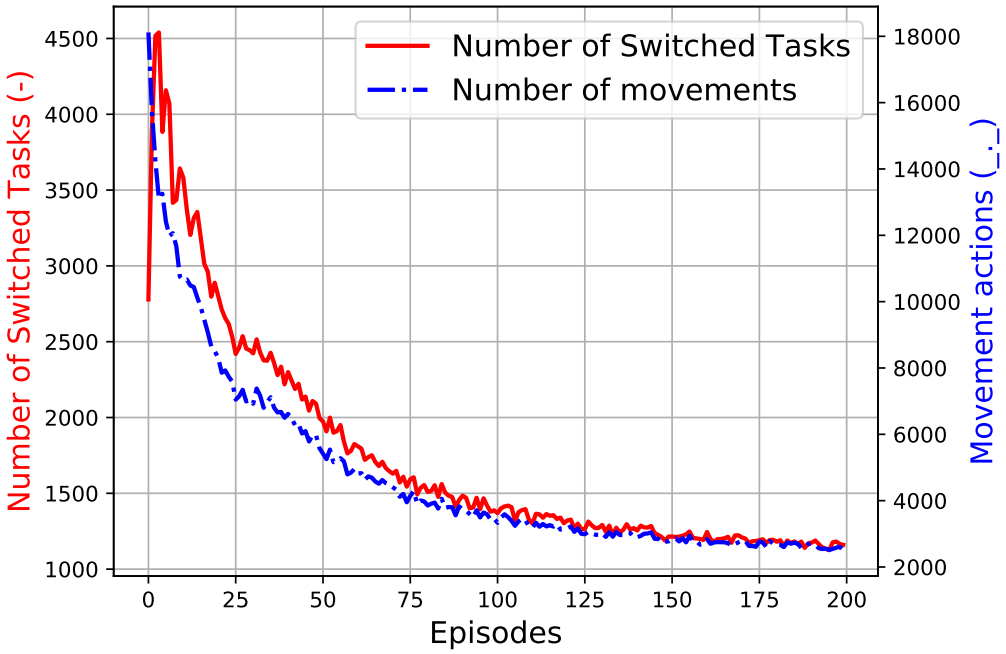

Fig. 4. Number of switched tasks and movements for all UAVs in each episode.

REFERENCES

[1] "Gartner says almost 3 million personal and commercial drones will be shipped in 2017," Feb 2017. [Online]. Available: http://www.gartner.com/newsroom/id/3602317

[2] M. Khaledi, A. Rovira-Sugranes, F. Afghah, and A. Razi, "On greedy routing in dynamic uav networks," in *2018 IEEE International Conference on Sensing, Communication and Networking (SECON Workshops)*, June 2018, pp. 1–5.

[3] S. Mousavi, F. Afghah, J. D. Ashdown, and K. Turck, "Leader-follower based coalition formation in large-scale uav networks, a quantum evolutionary approach," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 882–887.

[4] S. Mousavi, F. Afghah, J. D. Ashdown, and K. Turck, "Use of a quantum genetic algorithm for coalition formation in large-scale uav networks," *Ad Hoc Networks*, vol. 87, pp. 26–36, 2019.

[5] H. Peng, A. Razi, F. Afghah, and J. Ashdown, "A unified framework for joint mobility prediction and object profiling of drones in uav networks," *Journal of Communications and Networks*, vol. 20, no. 5, pp. 434–442, 2018.

[6] A. Rovira-Sugranes, F. Afghah, and A. Razi, "Optimized compression policy for flying ad hoc networks," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–2.

[7] A. Shamsoshoara, "Ring oscillator and its application as physical unclonable function (puf) for password management," *arXiv preprint arXiv:1901.06733*, 2019.

[8] A. Shamsoshoara, "Overview of blakley's secret sharing scheme," *arXiv preprint arXiv:1901.02802*, 2019.

[9] I. Stanojev, O. Simeone, Y. Bar-Ness, and T. Yu, "Spectrum leasing via distributed cooperation in cognitive radio," in *2008 IEEE International Conference on Communications*, May 2008, pp. 3427–3431.

[10] F. Afghah, M. Costa, A. Razi, A. Abedi, and A. Ephremides, "A reputation-based stackelberg game approach for spectrum sharing with cognitive cooperation," in *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, Dec 2013, pp. 3287–3292.

[11] N. Namvar and F. Afghah, "Spectrum sharing in cooperative cognitive radio networks: A matching game framework," in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*, March 2015, pp. 1–5.

[12] A. Valehi and A. Razi, "Maximizing energy efficiency of cognitive wireless sensor networks with constrained age of information," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 643–654, 2017.

[13] A. Valehi and A. Razi, "An online learning method to maximize energy efficiency of cognitive sensor networks," *IEEE Communications Letters*, vol. 22, no. 5, pp. 1050–1053, 2018.

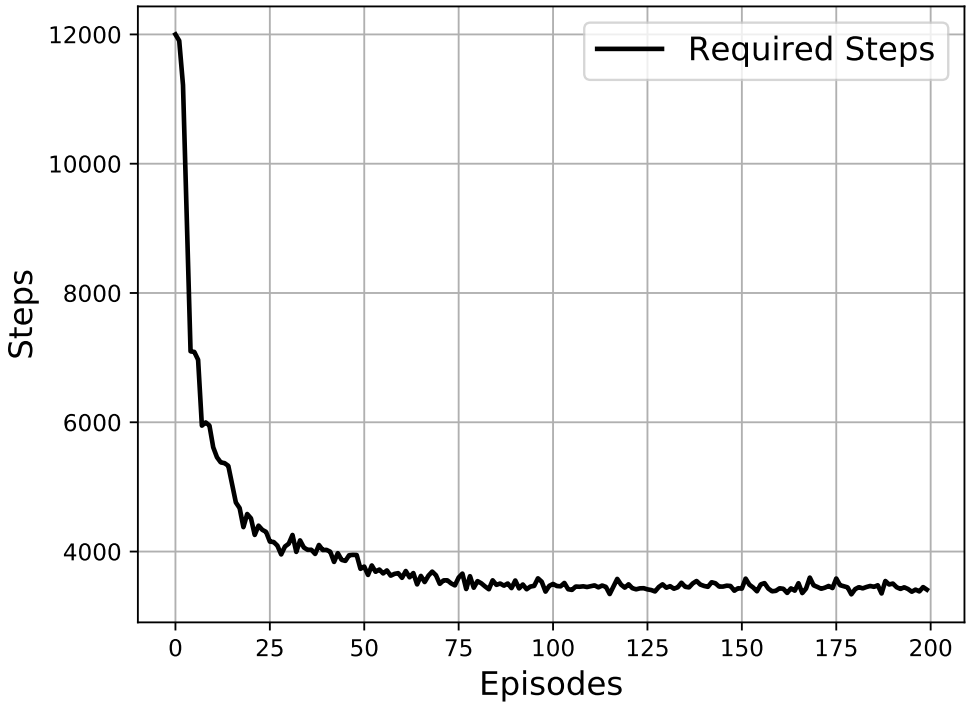

Fig. 5. Number of required steps in each episode to get 90% of the throughput summation.

[14] A. Razi, A. Valehi, and E. Bentley, "Delay minimization by adaptive framing policy in cognitive sensor networks," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[15] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337.

[16] J. R. Kok and N. Vlassis, "Collaborative multiagent reinforcement learning by payoff propagation," *Journal of Machine Learning Research*, vol. 7, no. Sep, pp. 1789–1828, 2006.

[17] A. Shamsoshoara, M. Khaledi, F. Afghah, A. Razi, and J. Ashdown, "Distributed cooperative spectrum sharing in uav networks using multi-agent reinforcement learning," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2019, pp. 1–6.

[18] F. Afghah, A. Shamsoshoara, L. Njilla, and C. Kamhoua, "A reputation-based stackelberg game model to enhance secrecy rate in spectrum leasing to selfish iot devices," *arXiv preprint arXiv:1802.05832*, 2018.

[19] A. Shamsoshoara and Y. Darmani, "Enhanced multi-route ad hoc on-demand distance vector routing," in *Electrical Engineering (ICEE), 2015 23rd Iranian Conference on*. IEEE, 2015, pp. 578–583.

[20] F. S. Mohammadi and A. Kwasinski, "Qoe-driven integrated heterogeneous traffic resource allocation based on cooperative learning for 5g cognitive radio networks," in *2018 IEEE 5G World Forum (5GWF)*. IEEE, 2018, pp. 244–249.

[21] M. Yu, J. Li, and H. Sadjadpour, "Amplify-forward and decode-forward: The impact of location and capacity contour," in *MILCOM 2005-2005 IEEE Military Communications Conference*. IEEE, 2005, pp. 1609–1615.

[22] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE communications Magazine*, vol. 42, no. 10, pp. 74–80, 2004.

[23] M. L. Littman, "Friend-or-foe q-learning in general-sum games," in *ICML*, vol. 1, 2001, pp. 322–328.

[24] Y.-C. Choi and H.-S. Ahn, "A survey on multi-agent reinforcement learning: Coordination problems," in *Proceedings of 2010 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*. IEEE, 2010, pp. 81–86.

[25] S. Rahili, B. Riviere, S. Oliver, and S.-J. Chung, "Optimal routing for autonomous taxis using distributed reinforcement learning," *IEEE*, 2018.

[26] S. S. Mousavi, M. Schukat, E. Howley, and P. Mannion, "Applying q (λ)-learning in deep reinforcement learning to play atari games," in *AAMAS Adaptive Learning Agents (ALA) Workshop*, 2017.

[27] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[28] S. Mousavi and F. Afghah, "Inter-and intra-patient ecg heartbeat classification for arrhythmia detection: a sequence to sequence deep learning approach," *arXiv preprint arXiv:1812.07421*, 2018.
[29] S. Mousavi, F. Afghah, and U. R. Acharya, "Sleepeegnet: Automated sleep stage scoring with sequence to sequence deep learning approach," *arXiv preprint arXiv:1903.02108*, 2019.
[30] S. Mousavi and F. Afghah, "Ecgnet: Learning where to attend for detection of atrial fibrillation with deep visual attention," *arXiv preprint arXiv:1812.07422*, 2018.
[31] S. S. Mousavi, M. Schukat, and E. Howley, "Traffic light control using deep policy-gradient and value-function-based reinforcement learning," *IET Intelligent Transport Systems*, vol. 11, no. 7, pp. 417–423, 2017.
[32] S. S. Mousavi, M. Schukat, and E. Howley, "Deep reinforcement learning: an overview," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2016, pp. 426–440.
[33] S. Mousavi, M. Schukat, E. Howley, A. Borji, and N. Mozayani, "Learning to predict where to look in interactive environments using deep recurrent q-learning," *arXiv preprint arXiv:1612.05753*, 2016.
[34] S. Mousavi, M. Schukat, and E. Howley, "Researching advanced deep learning methodologies in combination with reinforcement learning techniques," *Master's thesis, National University of Ireland Galway*, 2018.