Reinforcement Learning Aided Secure UAV Communications against Roaming Adversaries

Gongchao Su, Mingjun Dai, Bin Chen, Xiaohui Lin
College of Electrical and Information Engineering,Shenzhen University
Email:{gcsu,mjdai,bchen,xhlin}@szu.edu.cn

Abstract—**The rapid integration of Unmanned Aerial Vehicle(UAV) with existing network infrastructure brings enormous benefits to users, however it also introduces vulnerabilities to information security. In this paper we investigate a UAV-aided secure communication system, where a UAV is deployed to transmit confidential information to a ground user. Specifically, a mobile eavesdropper is moving in the vicinity of the ground user, attempting to intercept legitimate data transmissions. Its unpredictable trajectory poses as an extra security risk to UAV data transmission. Driven by this security challenge, we aim to optimize the UAV trajectory to maximize user average secrecy rate. Due to the fast changing environment caused by unpredictable movement of the eavesdropper, this nonconvex optimization problem is difficult to solve. Instead, we propose an online algorithm leveraging the Q-learning framework to deliver online decisions on UAV trajectory. With the help of carefully designed reward signals, the agent is able to learn an effective policy with desirable learning outcomes. Numerical results validate the effectiveness of the proposed algorithm, and shed light on learning outcomes with a variety of learning parameters.**

Index Terms—**physical layer security, unmanned aerial vehicle,Q-learning, reward signals, secrecy rate**

I. Introduction

Unmanned Aerial Vehicles(UAVs) promise great Line-of-Sight(LoS) connectivity, flexible deployment and coverage extension beyond traditional terrestrial access points(APs), and their integration into existing network infrastructure is essential to provide a seamless experience for network users. In disaster hit areas where sudden network failure occurs, UAVs can be quickly deployed to provide wireless access to network users. In hotspot areas where network service provided by existing terrestrial infrastructure can not meet user demand, UAVs can improve network capacity and mitigate network congestion. In addition, the versatility of UAVs is proved to be a greatly useful asset, as UAVs can act as flying transmitters/receivers,jammers,relays,etc. These multi-faceted benefits provided by UAVs is becoming a driving force for network paradigm shift toward ubiquitous connectivity [1].

Despite the numerous benefits, the forthcoming integration of UAVs with existing network infrastructure also brings some vulnerabilities. In particular, as wireless channels are in essence broadcast channels and LoS propagation can be easily exploited by a malicious attacker to enhance its eavesdropping ability, data transmission between UAVs and legitimate users are prone to eavesdropping attacks. Thus how to ensure data integrity and preserve user privacy becomes critical to provide security for users. Although higher level data encryption can be adopted to mitigate security risks, it consumes a lot of computing resources, which are relatively scarce in UAVs. Recently Physical Layer Security(PLS) has emerged as a promising method due to its ability to provide perfect secrecy regardless of computation resources and security protocols. Thus the application of PLS in UAV-aided communications has attracted significant research attentions [2].

The research was jointly supported by research grants from Natural Science Foundation of China (62071304), Natural Science Foundation of Guangdong (2020A1515010381),Basic Research foundation of Shenzhen City (20220809155455002,20200826152915001), and Natural Science Foundation of Shenzhen University (00002501).

To fully utilize the potential of UAV deployment, UAV trajectory must be optimized alongside other communication parameters, such as power and scheduling. This can be done via the traditional joint optimization method [2],by decomposing the main problem into several subproblems and employing an alternating iterative algorithm. However this approach require perfect information on the environment, which is difficult to obtain in dynamically changing scenarios. In addition, as high-dimensional optimization variables become increasingly large, this nonconvex optimization problem becomes increasingly intractable, and is difficult to converge and extremely computationally intensive. On the other hand, machine learning method can deliver online solutions to highly complex problems in a highly dynamic environment, and has attracted much recent attentions. In [3] a Reinforcement Learning(RL) algorithm is introduced to determine UAV trajectory to maximize user sum rates. In [4] a multi-agent deep RL algorithm is proposed to jointly optimized trajectory and power of UAVs and jammers. In [5] a deep Q network is proposed to solve the positioning and power of static UAVs. In [6] a deep RL algorithm is employed to optimize secrecy throughput in a UAV-aided NOMA communications network.

In this paper we propose a reinforcement learning algorithm to deliver online decision on UAV trajectory to optimize network secrecy throughput. Our work differs from other existing work in that we introduce more randomness in the considered scenario, by assuming eavesdroppers are mobile and their mobility pattern can not be predicted. To deal with the challenge of roaming attackers, we propose a Q-learning based online algorithm to optimize UAV trajectory, which requires no information on movement pattern of attackers. With the help of carefully designated reward functions, the agent is trained to complete its flight task and provide secure communications to users. Simulation results show the UAV agent can effectively learn to adapt its trajectory to roaming attackers.

II. System Model and Problem Formulation

A. System Model

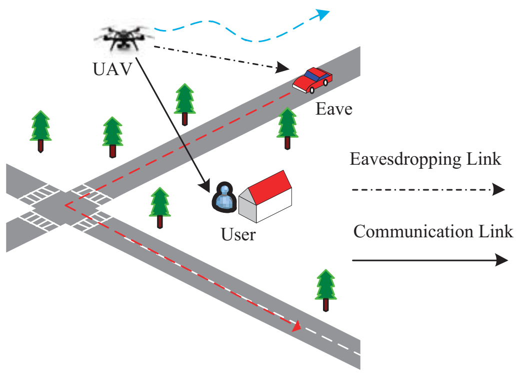

Fig. 1. An illustration of UAV secure communication system with moving adversaries

As depicted in Fig.1, we consider a UAV-aided communication network that consists of one UAV transmitter,one ground user(GU), and one roaming ground eavesdropper(eave). We consider a three dimensional Cartesian coordinate system and without loss of generality we assume the ground user is placed at the origin. During the flying time T, the UAV flys at a given altitude H from a predetermined starting point $\{x_s, y_s, H\}$, and end its mission at a landing point $\{x_f, y_f, H\}$. At the same time, a roaming eave is wandering in the vicinity of GU. When UAV and eave change their positions, they move at a maximum velocity of V_u and V_e,respectively. The flying time T is discretized into N time slots with equal length $\triangle t = \frac{T}{N}$, and at tth time slot, the UAV is located at the coordinate $\{x[t], y[t], H\}$,$t \in \mathcal{N} = \{1, 2, 3, \ldots, N\}$, and the eave is located at $\{x_e[t], y_e[t], 0\}$. Thus the UAV-GU distance $d_u[t]$ and UAV-eave distance $d_e[t]$ are given by

$$d_u[t] = \sqrt{x[t]^2 + y[t]^2 + H^2} \tag{1}$$

$$d_e[t] = \sqrt{(x[t] - x_e[t])^2 + (y[t] - y_e[t])^2 + H^2} \tag{2}$$

We assume the UAV is transmitting at a constant power P and the wireless link between UAV and GU is LoS dominated. Thus the channel gain $h_u[t]$ of the UAV-GU link and the channel gain $h_e[t]$ of the UAV-eave link are given by

$$h_u[t] = \frac{\beta_0}{d_u[t]^2} \tag{3}$$

$$h_e[t] = \frac{\beta_0}{d_e[t]^2} \tag{4}$$

where β_0 is the channel power gain at the reference distance of 1m. Hence the UAV-GU data transmission rate $R_u[t]$, and the UAV-eaves eavesdropping rate $R_e[t]$, are given by

$$R_u[t] = \log(1 + \frac{Ph_u[t]}{\sigma^2}) \tag{5}$$

$$R_e[t] = \log(1 + \frac{Ph_e[t]}{\sigma^2}) \tag{6}$$

where δ^2 is the thermal noise power. By exploiting the PLS mechanism, the UAV can provide an achievable secrecy rate R_{sec}, which is the rate difference between the legitimate UAV-GU link and the UAV-eave wiretap link

$$R_{sec}[t] = [R_u[t] - R_e[t]]^+ \tag{7}$$

where $[x]^+ = \max\{x, 0\}$.

B. Problem Formulation

Our goal is to maximize the average secrecy rate R_{sec}^{avg} throughout the entire flying period T. R_{sec}^{avg} is expressed as

$$R_{sec}^{avg} = \frac{1}{N}\sum_{t=1}^{N} R_{sec}[t] \tag{8}$$

Thus the secrecy rate optimization problem can be formulated as

$$\max_{\{x[t],y[t]\}} \quad R_{sec}^{avg} \tag{9}$$

$$s.t. \quad x[1] = x_s, y[1] = y_s \tag{9a}$$

$$x[N] = x_f, y[N] = y_f \tag{9b}$$

$$\sqrt{(x[t+1] - x[t])^2 + (y[t+1] - y[t])^2)} \leq V_u, t = 1, 2, \ldots, N-1 \tag{9c}$$

Several issues arise with problem (9). First, it is nonconvex since $R_{sec}[t]$ is the difference of two convex functions. Although its solution can be approximated using the Successive Convex Approximation(SCA) method [2], this method is generally slow to converge and computationally intensive,particularly with large dimensions of trajectory variables. Secondly, to solve this optimization problem, the path of eave movement $\{x_e[t], y_e[t]\}$ must be known in advance,which is impossible to predict for randomly moving adversaries, and this optimization problem has to be solved in a centralized and offline way. Hence it is desirable to make online trajectory decisions that can adapt to fast changing environment,i.e. randomly moving adversaries, and we leverage the Q-learning reinforcement learning method to maximize user average secrecy throughput R_{sec}^{avg}.

III. Q-Learning for Trajectory Design

Q-learning is a model-free,table-based reinforcement learning method that allows agents to learn through interactions with its environment [7]. The learning framework can be represented by a finite Markov Decision Process(MDP) that provides a formalization of sequential decision making concerning both immediate awards and future rewards. To allow agents to repeatedly interact with the environment in a sequence of time steps, we provide a finite MDP representation of agents and environment in the following subsection.

A. the MDP representation

In the proposed MDP framework, the flying UAV is the learning agent, and the GU and eave constitute its environment. At each time step $t \in \mathcal{N}$, the agent receives some representations of its environment's state, $s_t \in \mathcal{S}$, and selects an action $a_t \in \mathcal{A}$. At time step $t+1$, the agent receives a numerical reward,$r_{t+1} \in \mathcal{R}$,and enters a new state $S_{t+1} \in \mathcal{S}$. The finite sets of states, actions and rewards($\mathcal{S}, \mathcal{A}, \mathcal{R}$) allow us to define a state transition probability

$$p\{s^{'}|s,a\} \triangleq Pr\{s_{t+1} = s^{'}|s_t = s, a_{t+1} = a\} \tag{10}$$

and a discounting factor $\gamma \in [0,1)$ that determines the present value of future rewards. Thus our MDP can be fully represented by the tuple $\{\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P}, \gamma\}$:

- State space $\mathcal{S}$. This state space $\mathcal{S} = \{s_1, s_2, \ldots\} \bigcup \{s_L\}$ is the representation of the environment, and we use the UAV and eave's ground locations to characterize $\mathcal{S}$. That is ,$s_n = \{x[n], y[n], x_e[n], y_e[n]\}, x[n] \neq x_f, y[n] \neq y_f$. The state s_L represents the landing state where the agent reaches the landing position, $s_L = \{x_f, y_f\}$. To make state space finite, we further discretize the horizonal plane into $\mathcal{K} \times \mathcal{K}$ grids, and we assume the UAV is located at those grid points at each time step. Similarly we assume the eave moves in the vicinity of GU and its movement area is also discretized into a smaller $\mathcal{M} \times \mathcal{M}$ grid world.
- Action space $\mathcal{A}$.Without loss of generality, we allow the UAV to move in four directions, or remain static. Thus $\mathcal{A} = \{'static', 'left', 'right', 'up', 'down'\}$.
- Reward function $\mathcal{R}$. The reward not only concerns the secrecy rate, but also involves some auxiliary awards that gives penalties to boundary violations and encourages the UAV to fly close to GU.
- State transition probability $\mathcal{P}$
- Discounting factor $\gamma \in [0,1)$

When the agent chooses some action $a_t \in \mathcal{A}$ at time step t, it moves in corresponding direction and this MDP transitions from state s_t to the next state s_{t+1}. However, to force the agent stay at the landing point when time limit expires, we use state s_L as the terminal state that ends one episode. This landing state can only transition to itself at the next time step regardless of whatever action the agent may take,i.e. $p\{s_L|s_L\} = 1$.

B. the Q-learning Process

The goal of Q-learning for agent is to learn a behavior rule,or a *policy* π,that determines the probability of selecting each possible action in a given state. If the agent is following policy π at time t, then

$$\pi(s,a) = Pr\{a_t = a|s_t = s\} \tag{11}$$

When the agent follows policy π in state s, s can be assigned a value through a state-action value function to reflect the expected future return. In Q-learning, this state-action value is represented by the Q values, that approximate future rewards regardless of policy followed after current time step

$$Q_\pi(s,a) = \mathbb{E}\{R_t|s_t = s, a_t = a\} \tag{12}$$

where R_t is the discounted cumulative future rewards at time step t

$$R_t = \sum_{k=0}^{T-1} \gamma^k r_{t+k+1} \tag{13}$$

Given state-action values $Q_\pi(s,a)$,a greedy policy can be employed to select the optimal action

$$\pi^*(s,a) = \arg\max_a Q^*(s,a) \tag{14}$$

where $Q^*(s,a)$ is the optimal value of state-action pair (s,a). This value can be approximated using an iterative updating rule [7]

$$\begin{aligned} &Q_\pi(s_t,a_t) \leftarrow Q_\pi(s_t,a_t) + \\ &\alpha(r_t + \gamma \max_a Q_\pi(s_{t+1},a) - Q_\pi(s_t,a_t)) \end{aligned} \tag{15}$$

where the last part of eq.(15) is the temporal-difference(TD), or the TD error between current estimate and updated estimate based on next state and next reward. The learning rate $\alpha \in [0,1]$ is a step-size parameter that determines the importance of new information on estimates and discounting factor $\gamma \in (0,1]$ determines the importance of future rewards with respect to immediate reward. This Q-learning process is guaranteed to converge under proper size α [7].

C. Q-Learning for UAV Trajectory Design Against Roaming Adversaries

In this section we proceed to design our Q-learning framework for UAV trajectory planning. Since the agent flies in a bounded area, a penalty r_{pen} of negative real number must be given to the agent to penalize violations of boundary conditions. When the agent takes any action that results in a boundary violation,it receives the penalty r_{pen} and reduces the value of the state-action pair, and remain static at next time step. Furthermore, when the agent flies in areas with zero secrecy rate,i.e., areas far away from users and with very poor link quality and rate,it should be encouraged to learn to fly close to users and try to find better places with positive secrecy rate. Hence we give an extra reward r_d to the agent, depending on the Euclidean distances between agent and GU. In addition, to direct the agent to learn to fly to the landing position, we designate a specific reward $r_l[t]$ that gradually increase its penalty to the agent as time step approaches the time limit. These rewards constitute the auxiliary reward r_{au} at time slot t as

$$\begin{aligned} r_{au}[t] &= r_{pen}[t] + r_d[t] + r_l[t] \\ &= r_{pen}[t] + \frac{\zeta}{d_u[t]} - \frac{\eta d_l[t]}{N+1-t} \end{aligned} \tag{16}$$

where ζ, η are hyperparameters that control the weight of distance-based rewards, and $d_l[t] = \sqrt{(x[t]-x_f)^2 + (y[t]-y_f)^2}$ is the distance to the landing location.

When agent reaches the landing point at the final time step, with $d_l[N] = 0$ the penalty $r_l[N] = 0$. Furthermore,the agent is given a one-time positive reward r_f, which is a relative large real number to encourage agent to stay at the landing point when flight time ends. The reward given to agent at time step t is a combination of secrecy rate,the auxiliary reward and the end point arrival reward

$$r_t = (R_{sec}[t] + r_{au}[t])(1 - I(t)) + r_f \tag{17}$$

where $I[t]$ is a 0-1 indicator and $I(t) = 1$ if agent reaches the landing point and 0 otherwise. It can be seen that when agent reaches the landing point, it receives the one-time reward r_f and zero reward afterwards before time expires.

At the start of training, to encourage agent to explore all possible actions, all table entries of Q values for state-action pairs are assigned a positive initial value q_0. The agent picks its optimal action from its current state s according to a ξ-greedy policy

$$a_t = \begin{cases} \arg\max\limits_a Q_t(s,a), \text{with } 1-\xi \text{ probability} \\ \quad \text{random action,with } \xi \text{ probability} \end{cases} \tag{18}$$

The parameter ξ determines whether the agent should stick to its current best action, or explore other possible actions, and is referred as the *exploration-exploitation* tradeoff. As training continues, ξ can be gradually decayed from a higher initial value ξ_0,i.e. $\xi_0 = 1$, to a lower bound ξ_f

$$\xi_t = \max\{\xi_f, \xi_t - \frac{2\xi_0}{\text{number of episodes}}\} \tag{19}$$

Our Q-learning algorithm is presented in Algorithm 1.

Algorithm 1 Q-learning Algorithm for UAV Trajectory Design Against Roaming Adversaries

```
 1: Initialization: State space $\mathcal{S}$,action space $\mathcal{A}$, Q-table
    $Q(s,a) \leftarrow q_0, \forall s \in \mathcal{S}, \forall a \in \mathcal{A}$, learning rate $\alpha$, discount-
    ing factor $\gamma$,greedy probability $\xi \leftarrow \xi_0$.
 2: for each episode do
 3:     Return both the UAV and the eave to their starting
        positions. Receive the initial observation $s$.
 4:     for $t = 1, 2, \ldots, N$ do
 5:         Choose action $a_t$ using $\xi$-greedy policy according
            to eq.(18).
 6:         Get reward $r_t$ according to eq.(17) and observe the
            next state $s_{t+1}$.
 7:         Update $Q(s_t, a_t)$ according to eq.(15).
 8:         Update state: $s_t \leftarrow s_{t+1}$.
 9:     end for
10:     Decay the greedy probability $\xi$ according to eq.(19).
11: end for
```

IV. Numerical Results

In this section we evaluate the performance of our proposed algorithm. We consider a 100m×100m horizonal plane and the UAV flies at a fixed height $H = 50$m. The horizonal plane is discretized into a 20×20 grid world with an axis interval $\triangle d =$ 5m. GU is located at the origin, and the UAV starts its flight at $\{-30, -30, 50\}$m and finishes its mission at $\{30, 30, 50\}$m. The eave starts its movement at $\{-5, -5, 0\}$m and follows a random walking model. The movement area of the eave is discretized into a smaller 5×5 grid world with $\triangle d = 5$m and centered around the origin at ground level. Time step is set as $\triangle t = 1$s. Other than remaining static, both the UAV and the eave move in four directions with speed $V_u = V_e = 5$m/s. The UAV transmits at a power level of $P = 30$dbm, and the channel gain at the reference point is set as $\beta_0 = -30$dB. We set the thermal noise power $\sigma^2 = -50$dbm. The flying time is set as $T = 200$s.

The learning hyperparameters are configured as follows. The learning rate is set as $\alpha = 0.3$, and the initial exploration ratio is $\xi_0 = 1$ and the lower bound is $\xi_f = 0.1$. The discounting factor is $\gamma = 0.99$. Each time the agent chooses an action that leads to boundary condition violations,agent remains static and receives a penalty $r_{pen} = -100$. The one-time reward for arrival at end location is $r_f = 3000$.

Fig.2 depicts the UAV trajectory after completing $n = 180000$ episodes of training with learning parameters $\alpha = 0.3, \gamma = 0.99$. At the early stage of its fight, the agent learns to fly close to the user, thanks to the increasing rewards given with shorter distances to the user. Then the agent tries to find positions in the vicinity of the user with better secrecy rate. In the final stage, when large distance to the landing position triggers increasingly large penalty as time approaches time limit, the agent learns to first fly close to, and then reach the landing location in order to lower the penalty and receive the relatively large landing reward. This shows that the agent is able to infer useful information on network topology, and with the help of reward signals, the agent is able to adopt a suitable policy with desirable learning outcomes.

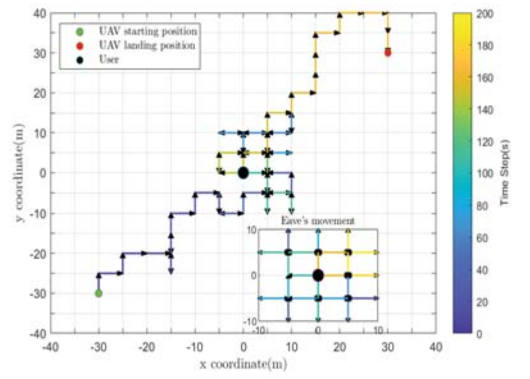

Fig. 2. UAV trajectory after $n = 180000$ episodes of training with learning parameters $\alpha = 0.3, \gamma = 0.99$ in the simulated environment of one user and one moving eavesdropper. The color bar shows time steps in different colors and movement of the eave is shown in the smaller axes.

Fig.4 shows the episodic rewards per training episodes over the entire flying period with different configurations of learning parameters. The agent is trained up to $n = 200000$ episodes in all three setups. Note that the plot shows the average episodic rewards per 1000 episodes. It can be seen that in all three setups the learning algorithm converges, and the agent is able to receive high episodic rewards by successfully reaching the terminal state s_L and thus receiving the large landing reward. One can see that with $\alpha = 0.01$ the agent is able to receive relatively higher episodic rewards in the later stage of training, compared with $\alpha = 0.1$ and $\alpha = 0.3$.

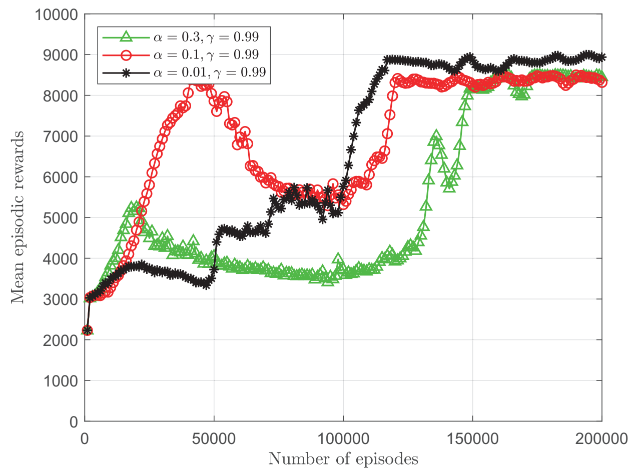

Fig. 3. Mean episodic rewards versus training episodes with different learning rates α. Rewards are averaged over 1000 episodes.

Fig.4 shows the average user secrecy rate versus training episodes with different learning parameter α. It can be seen that when training ends the agent achieves a relatively stable user secrecy rate in all three setups. One can observe that with $\alpha = 0.01$ the agent is able to achieve higher average secrecy rate. Since there is no explicit rule to choose the learning parameters, $\alpha = 0.01$ in combination with $\gamma = 0.99$ can produce relatively good results.

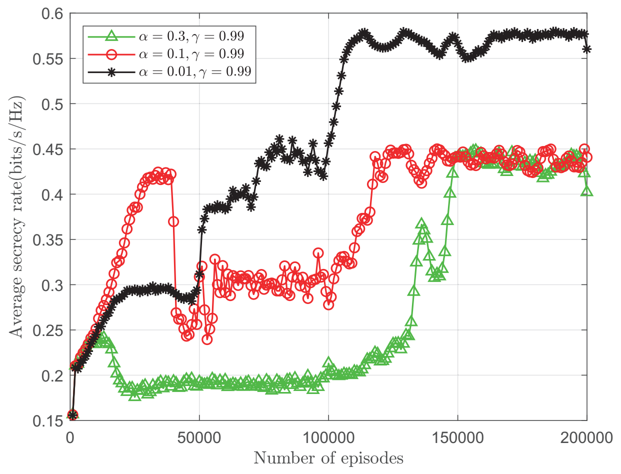

Fig. 4. Achievable average secrecy rate versus training episodes with different learning rates α. Secrecy rates are averaged over 1000 episodes.

V. Conclusion

In this paper we have investigated UAV trajectory design to provide secure data transmissions for ground users in the presence of moving eavesdroppers. The unpredictable movement of roaming eavesdroppers bring more randomness and introduce extra security risks, and call for an online solution to optimize user secrecy rate. To deal with this challenge we have proposed a Q-learning based online algorithm to determine UAV trajectory. With the help of carefully designed reward signals the agent is shown to effectively learn to find a trajectory that optimize user secrecy rate and complete its fight task from its starting position to its landing location. Simulation results have validated the effectiveness of our algorithm, and provided useful insight on learning outcomes with a diverse configuration of learning parameters.

Our work have primarily focused on the classical tabular Q-learning method, which does not scale well with higher dimensions of state and action spaces. In future work we will consider more state-of-the-art Q-learning variants, and combine function approximation methods and deep neural networks with Q-learning, and consider multiple UAVs as learning agents or adversaries.

References

[1] G. Geraci, A. Garcia-Rodriguez, M. M. Azari, A. Lozano, M. Mezzavilla, S. Chatzinotas, Y. Chen, S. Rangan, and M. D. Renzo, “What will the future of uav cellular communications be? a flight from 5g to 6g,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1304–1335, 2022.

[2] J. Wang, X. Wang, R. Gao, C. Lei, W. Feng, N. Ge, S. Jin, and T. Q. S. Quek, “Physical layer security for uav communications: A comprehensive survey,” *China Communications*, vol. 19, no. 9, pp. 77–115, 2022.

[3] H. Bayerlein, P. De Kerret, and D. Gesbert, “Trajectory optimization for autonomous flying base station via reinforcement learning,” in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.

[4] Y. Zhang, Z. Mou, F. Gao, J. Jiang, R. Ding, and Z. Han, “Uav-enabled secure communications by multi-agent deep reinforcement learning,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11 599–11 611, 2020.

[5] A. S. Abdalla, A. Behfarnia, and V. Marojevic, “Aerial base station positioning and power control for securing communications: A deep q-network approach,” in *2022 IEEE Wireless Communications and Networking Conference (WCNC)*, 2022, pp. 2470–2475.

[6] L. P. Qian, W. Zhang, H. Zhang, Y. Wu, and X. Yang, “Secrecy capacity maximization for uav aided noma communication networks,” in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 3130–3135.

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