Towards Deep Q-Learning for Target k-Coverage Protocol In UAV Networks

Manel Chenait*, Mohammed Anis Guermat*, Meriem L'Atra Mizat*, Ala' Khalifeh §

* University of Sciences and Technology Houari Boumediene (USTHB), Algiers, Algeria
§ German Jordanian University (GJU), Amman, Jordan
Emails: manel.chenait@usthb.edu.dz, mohamedanis.guermat@etu.usthb.dz
meriemlatra.mizat@etu.usthb.dz, ala.khalifeh@gju.edu.jo

***Abstract*—Unmanned Aerial Vehicles (UAVs or drones) play a crucial role in surveillance missions, especially where ground infrastructure is damaged or inaccessible. Ensuring reliable and simultaneous coverage of critical zones by UAVs, known as k-coverage, remains a significant challenge. Traditional methods require UAVs to cover an entire area which leads to high energy consumption, this is problematic, especially in environments where battery recharge or replacement is difficult. To overcome these challenges, Only a set of targets should be monitored instead of monitoring the entire area. This paper proposes DQTCP (Deep Q-learning-based Target Coverage Protocol), a new deep reinforcement learning approach to continually cover a maximum number of stationary targets. In DQTCP, the UAV acts as an autonomous Deep Q-Network (DQN) agent, with discrete actions and individualized learning parameters balancing exploration and exploitation. Through iterative training and environment interaction, UAV adopts policies that optimize the target coverage effectiveness. Simulations show that DQTCP using based on the Reinforcement learning theory, is very efficient in terms of coverage performance and stability.**

***Index Terms*—Target k-Coverage, UAVs, Reinforcement Learning, DQN.**

I. Introduction

Unmanned Aerial Vehicles (UAVs) become indispensable tools for surveillance and monitoring in environments where ground infrastructure is damaged, inaccessible, or insufficient. They provide critical aerial capabilities to collect real-time data and imagery in areas that are difficult or dangerous to reach by conventional means. In fact, a fundamental challenge in UAV surveillance applications is to ensure a reliable coverage strategy where each critical zone within the area of interest must be simultaneously monitored by k UAVs (k-coverage) [1]–[5]. However, in this case, each UAV must traverse the entire area and simultaneously ensure that every point is covered by multiple UAVs, which results in a significantly high energy consumption cost, especially in hostile environments where recharging or replacing UAVs' batteries is difficult or impossible. Indeed, there is an urgent need to develop a new monitoring approach that focuses on selective monitoring of critical targets, rather than continuous monitoring of the entire vast area. To ensure that each target should be covered by several UAVs. In this paper, we propose DQTCP (*Deep Q-learning-based Target Coverage Protocol*), a new coverage protocol allowing an UAV to achieve energy-aware k-coverage of stationary ground targets through deep reinforcement learning. DQTCP defines the environment, UAV configuration, and coverage constraints to ensure that each target is surely covered by the UAV. This latter functions as an independent Deep Q-Network (DQN) agent with a discrete action space that includes directional movements and a stationary action. Learning parameters such as exploration rate, learning rate, and discount factor regulate the balance between exploration and exploitation. During training, UAVs iteratively interact with the environment, selecting actions using an epsilon-greedy policy and receiving feedback based on coverage effectiveness.

The remainder of this paper is organized as follows: Section II discusses related work. Section III explains Q-learning and Deep Q-learning principles. Section IV describes the proposed algorithm *DQTCP*. Section V provides simulation results. Finally, Section VI concludes the paper.

II. Related Work

This section discusses the most representative k-coverage protocols in Multi-UAV networks, along with a summary of their shortcomings.

The authors in [6] propose a multi-agent reinforcement learning framework to optimize the coverage area of multiple (UAVs) while minimizing overlaps between their observation zones. Each UAV operates as an autonomous agent that learns to effectively cover its assigned environment. A coordination mechanism is triggered when UAVs are in proximity, allowing them to exchange local Q-values and achieve an equilibrium in their joint action selection. The environment is modeled as a three-dimensional grid, and the reward structure is designed to penalize redundant

coverage and encourage movement stability as well as exploration of previously uncovered regions. The proposed approach is validated in a simulated scenario involving two UAV agents, demonstrating that the UAVs successfully maximize coverage with minimal overlap. Nonetheless, the current study is limited to a small number of agents and a discrete environment, which constrains the scalability and applicability of the method in more complex and realistic operational settings.

The authors in [1] designed an algorithm for target coverage in wireless sensor networks, based on Q-learning. The system is centralized and aims to extend network lifetime while ensuring efficient target coverage. The process is modeled as a MDP (Markov Decision Process), where an intelligent agent selects which sensors to activate based on a reward function integrating remaining energy and number of targets covered. Sensors are placed randomly in a square area, with fixed detection radii, and are progressively activated according to the algorithm's decisions. However, the model is limited to fixed sensors and static environments, with no possibility of mobility or distributed coordination.

Sun et al. [3] propose a new method for persistent area coverage using UAVs based on deep reinforcement learning, incorporating a cooperative reward mechanism . The system is implemented in a distributed manner, where each UAV leverages a shared bidirectional recurrent neural network (BRNN) to coordinate its actions. The primary objective is to minimize the age of each cell, defined as the time elapsed since its last visitation, this operation enables continuous and balanced coverage over the monitored area. The approach supports dynamic mission scenarios, allowing UAVs to join or leave the operation without compromising the overall coordination strategy. Simulation results indicate that the proposed protocol decreases the average cell age compared to traditional methods. However, the reliance on a fully connected communication network among UAVs and the lack of real-world experiments restrict the comprehensive validation of the proposed solution.

Zhang et al [4] conduct an incremental investigation into area coverage using UAVs, beginning with a single agent and extending to a multi-agent framework. Initially, tabular Q-learning is applied within discrete environments, followed by a graph-based modeling approach to address irregular spatial configurations. Subsequently, a deep reinforcement learning technique, Actor-Critic with Trust Region (ACKTR), is employed for two UAVs to optimize coverage efficiency. The objective is to maximize area coverage while minimizing overlap and the number of movement steps. Additionally, the method incorporates hexagonal tessellation to enhance spatial coverage uniformity. Although the approach is methodically developed and demonstrates technical robustness, it is limited by the omission of energy consumption considerations, obstacle avoidance, and explicit inter-UAV coordination.

Jian Xiao et al [5] propose a dynamic coverage algorithm for UAVs operating in complex environments, leveraging deep reinforcement learning under the Fusion-based Discrete Soft Actor-Critic (FDSAC) framework. The method allows each UAV to independently make decisions (e.g., optimal path, obstacle avoidance, parameter adjustment) based on local observations while benefiting from training on the global state. The environment is modeled as a grid incorporating obstacles, control noise, and communication constraints. Experimental results demonstrate that FDSAC achieves fast and resilient area coverage. Nonetheless, the approach remains constrained to discrete grid environments.

III. Q-learning and Deep Q-learning Principle

Q-learning is a foundational reinforcement learning algorithm that enables an agent to learn optimal actions in a given environment through trial and error, by estimating the expected rewards of state-action pairs. It functions by iteratively updating a Q-value function using observed rewards and future value estimates, thereby guiding the agent toward maximized cumulative rewards. However, traditional Q-learning relies on tabular methods which become inefficient or infeasible with large or continuous state and action spaces.

Deep Q-learning overcomes these limitations by employing deep neural networks to approximate the Q-value function, enabling reinforcement learning to scale to complex, high-dimensional problems. By integrating techniques such as experience replay and target networks, Deep Q-learning stabilizes training and improves performance in dynamic environments. This section explores the principles behind both Q-learning and its deep learning extension, highlighting their mechanisms, strengths, and applications in decision-making problems. Q-learning estimates the quality, or Q-value, $Q(s, a)$, which represents the expected cumulative reward for taking action a in state s.

The Q-values are updated iteratively using:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right]$$

where s is the current state, a is the action taken, r is the immediate reward, s' is the next state, α is the learning rate, and γ is the discount factor for future rewards.

By applying this update rule over time, the agent learns to choose actions that maximize long-term rewards. This process balances exploration of new actions and exploitation of known rewarding actions, ultimately converging to an optimal policy.

Deep Q-Networks (DQNs) are a class of reinforcement learning algorithms that operate within the framework of Markov Decision Processes (MDPs). An MDP provides a formal model for decision- making where an agent interacts with a stochastic environment characterized by states, actions, transition probabilities, and rewards.

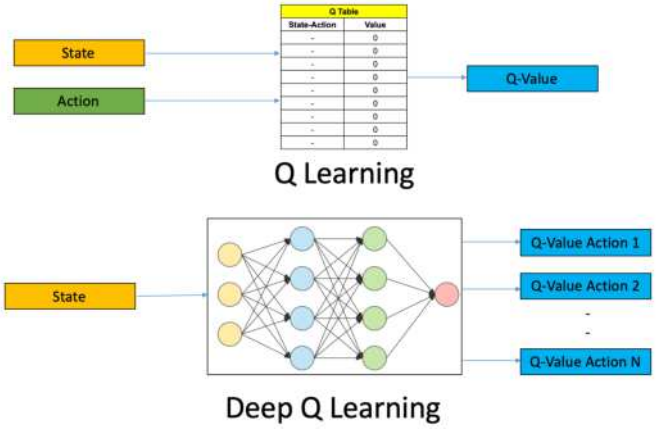

Fig. 1: DQN Principle

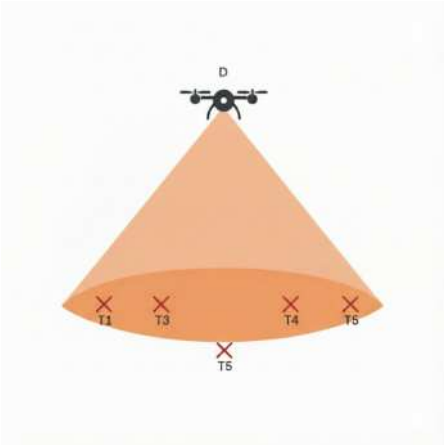

Fig. 2: One UAV Cover Multiple Targets.

Q-learning, the foundation of DQN, aims to find the optimal action-value function $Q(s, a)$ in an MDP without requiring explicit knowledge of the environment's dynamics. The Q-function represents the expected return of taking action a in state s and following the optimal policy thereafter. DQN extends traditional Q-learning by approximating the Q-function with a deep neural network parameterized by θ, enabling it to handle large or continuous state spaces. The network is trained using experience replay and target networks to stabilize learning.

IV. DQTCP: Deep Q-Learning For Target k-Coverage Protocol

Based on the DQN principle, DQTCP (Deep Q-Learning For Target k-Coverage Protocol) achieves k-coverage for a set T composed of m static targets scattered across the area. This requires that each target be concurrently monitored by a minimum of k UAVs, thereby improving the system's reliability and fault tolerance for the surveillance task. Specifically, the UAV operates as an autonomous DQN agent that interacts with its environment and incrementally learns to achieve the required coverage degree k. The capacity of DQN to generalize over high-dimensional state spaces allows efficient learning and adaptation despite the continuous and evolving nature of the UAV position, resources, and spatial constraints.

The indicator function $\delta(D, t_j)$ determines whether the UAV $'D'$ can cover the target t_j or not:

$$\delta(D, t_j) = \begin{cases} 1, & \text{If } d(D, T_j) \leq R \\ 0, & \text{otherwise} \end{cases}$$

Where R is the radius of the UAV and $d(D, T_j)$ is the distance between a UAV and a target T_j :

$$\mathrm{d(D,\ T_j)} = \sqrt{(x_D - x_j)^2 + (y_D - y_j)^2}$$

Fig. 3: UAV Actions.

The UAV can perform six actions: *up*, *down*, *left*, *right*, *scan*, and *stay*. *Up* moves the UAV one unit upward on the grid, while *down* moves it one unit downward. The actions *left* and *right* move the UAV one unit to the left and right, respectively. The *scan* action allows the UAV to detect targets and update the local coverage degree k. Finally, the *stay* action keeps the UAV stationary in its current position (Fig3).

TABLE I: DQTCP Parameters

$T_1, T_2, \ldots, T_m$	m Targets
Pos_D	UAV's Position
$PosT_j$	Target's Position j ($j <= m$)
R	Sensing Radius
k	Coverage degree
Actions	{*Up, Down, Left, Right, Scan, Stay*}
s	*State*
ϵ	Exploration rate
θ	Random weight
γ	Indicator function

The DQTCP environment is defined by the initial UAV position $PosD$, the target positions $PosT_j$, and the UAV actions (*up*, *down*, *left*, *right*, *scan*, *stay*) (line 1). The UAV is modeled as an autonomous agent characterized by its

Algorithm 1 DQTCP Algorithm

1: **Init the environment parameters** :
2: m, $PosD$, $PosT_j$, k, $actions$
3: **Init the DQTCP agent parameters** :
4: s, ϵ, θ , γ
5: **for** episode $= 1$ **to** $max\_episodes$ **do**
6: $total\_reward \leftarrow 0$
7: **for** etape $= 1$ **to** max_steps **do**
8: **if** random $< \varepsilon_i$ **then**
9: $a_i \leftarrow$ random action ▷ exploration
10: **else**
11: $a_i \leftarrow \arg\max_a Q_i(s, a; \theta)$ ▷ exploitation
12: **end if**
13: Observe the new state s' then assess the situation
14: $On\_Target()$?
15: $How\_Cov()$?
16: Award the prize r_i according to the rules of coverage and interference
17: Update
18: $y_t = R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \theta)$
19: $\mathcal{L}(\theta) = (y_t - Q(s_t, a_t; \theta))^2$
20: **end for**
21: $s \leftarrow s'$ ▷ update of the current status
22: r= CalculateReward()
23: **end for**
24: $total\_reward \leftarrow total\_reward + \sum_i r_i$
25: Reduce ε_i
26:

state vector s, exploration probability ϵ, deep Q-network parameters θ and the discount factor γ. At the beginning of the training, all parameters are initialized then the learning process proceeds over a predefined number of episodes (line 4), each episode consists of a sequence of interaction steps during which the UAVs repeatedly observe the environment, take actions, and receive feedback.

At every decision step of an episode, the UAV D selects an action a_i according to an ϵ-greedy policy to balance exploration and exploitation. With probability ϵ_i, the agent chooses a random action from the discrete action space to explore unvisited or less-visited state–action pairs, whereas with probability $1-\epsilon_i$ it exploits its current knowledge by selecting the action that maximizes the estimated state-action value, i.e., $a_i = \arg\max_a Q_i(s, a; \theta)$ (lines 5-13). Once the selected actions are executed, the system transits to a new state s'. Specific conditions are evaluated through boolean functions such as `On_Target()` (line 14), which checks if the UAV is located on or near a target, `How_Cov()` (line 15), which verifies If the drone has covered an acceptable number of targets. These evaluations are used to compute an instantaneous reward r_i with positive rewards assigned when the UAV contributes to cover more than one target and negative rewards assigned otherwise (line 17). Learning in DQTCP is driven by temporal-difference updates of the deep Q-network. For each transition $(s_t, a_t, r_{t+1}, s_{t+1})$, a target value y_t is computed as (line 19):

$$y_t = R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \theta),$$

which combines the immediate reward with the discounted estimate of the optimal future return from the next state. The Q-network parameters θ are updated by minimizing the squared temporal-difference error (line 20)

$$\mathcal{L}(\theta) = \big(y_t - Q(s_t, a_t; \theta)\big)^2,$$

Across episodes, the exploration rate ϵ_i is gradually decreased according to a predefined schedule, so that the policy evolves from exploratory behavior in the early stages to more deterministic. At the end of each episode, the $total\_reward$ is computed as the sum of all rewards (line 24).

V. Performance evaluation

Experiments of DQTCP are conducted using Python 3.10.13 under Windows 10 with its many libraries including Gymnasium 0.28.1 for custom reinforcement learning environment construction, NumPy 1.26.4 for numerical computation, TensorFlow 2.10.0 and Keras 2.10.0 for Deep Q-Network implementation, and Pygame 2.6.1 for environment visualization [7]. The simulation parameters are summarized in Table II.

The UAVs start at a predefined position and move in discrete actions (up, down, left, right, stay, scan). The reward function encourages configurations in which most targets are covered by the UAV. The task ends after a fixed number of steps.

This environment models a scenario in which an UAV navigates in a 2D space to reach one or multiple target as illustrated in Fig 4. At each stage, the drone chooses a discrete movement (up, down, left, right, or stay in place) based on its position and the location of the target. A reward is only given when the drone reaches a target, and increases when the target is reached. The task ends either when all targets have been covered or when a fixed number of stages has been exceeded.

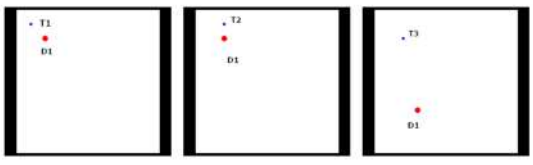

Fig. 4: DQTCP Grid at discrete times

The blue dot represents the current position of the target, while the red one indicates the drone.

DQTCP is evaluated in terms of the following metrics:

TABLE II: Simulation Parameters.

Simulation			
	Topography	Structural Arrangement	20m x 20m
	n	Number of UAVs	1
	m	Number of Targets	5
	Max Steps	Maximum steps allowed per episode	30
Rewards and Penalties	**Parameter**	**Details**	**Value**
	Coverage Reward	Reward for simple target coverage	+10
	k-Coverage Reward	Reward for multiple target coverage	+20
	No Coverage Penalty	Penalty for targets left uncovered	-10

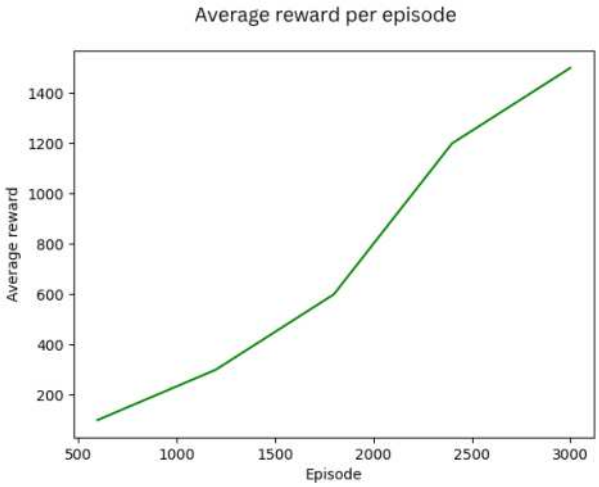

Fig. 5: Average Reward Per Episode of a Single DQTCP Agent.

- The average reward of a single DQTCP agent;
- k-coverage Achievement.

A. The average reward of a single DQTCP agent

Fig 5 presents the evolution of the average reward attained by a single DQTCP agent across training episodes. The observed plot demonstrates a consistent and gradual increase in reward values, reflecting a systematic improvement in the agent's policy as it accumulates experience. From episode 3000 onward, the reward curve achieves stability and attains the maximum value, indicating convergence to an optimal and stable policy within the given training environment. This is due to the simplified environmental structure characterized by well-defined objectives and a limited state-action space. In fact, the exploration and the exploitation are balanced to optimize the total reward.

B. k-coverage Achievement

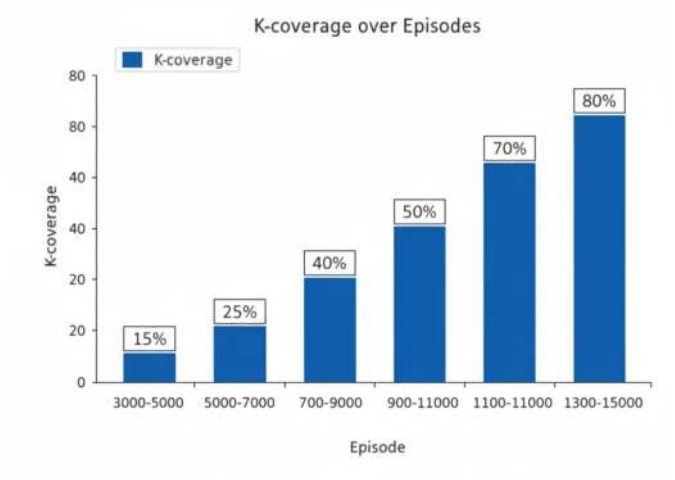

Fig. 6: K-coverage achievement over episode ($k = 2$)

Fig. 6 shows that average coverage when multiple UAV are deployed . Starting at about 23% and reaching 80% after 15000 episodes. Since the curve does not level off, the agents have not yet converged to an optimal strategy. Instead, they keep improving their learning and coordination to handle challenges like multi-agent cooperation, autonomy limits, collision avoidance, and restricted zones.

VI. Conclusion

This paper presents DQTCP, a new Deep Q-Learning protocol that enables efficient k-coverage for autonomous UAV swarms. The UAV operates as a Deep Q-Network agent, learning to cooperatively cover fixed targets while maximizing the number of covered targets at one time. Simulations validate the protocol's effectiveness and robustness, paving the way for real-world UAV deployments. Future enhancements will integrate multi-UAVs collaboration to achieve the k-coverage of the maximum number of targets. Furthermore, we will introduce explicit energy models, including battery consumption and recharging constraints, to prolong missions in UAVs environments.

References

[1] P. Xiong, D. He, and T. Lu, "A q-learning based target coverage algorithm for wireless sensor networks," *Mathematics*, vol. 13, no. 3, p. 532, 2025.

[2] J. Ni, Y. Gu, Y. Gu, Y. Zhao, and P. Shi, "Uav coverage path planning with limited battery energy based on improved deep double q-network," *International Journal of Control, Automation and Systems*, vol. 22, no. 8, pp. 2591–2601, 2024.
[3] Z. Sun, N. Wang, H. Lin, and X. Zhou, "Persistent coverage of uavs based on deep reinforcement learning with wonderful life utility," *Neurocomputing*, vol. 521, pp. 137–145, 2023.
[4] C. Zhang, "Area coverage with unmanned aerial vehicles using reinforcement learning," 2020.
[5] J. Xiao, G. Yuan, Y. Xue, J. He, Y. Wang, Y. Zou, and Z. Wang, "A deep reinforcement learning based distributed multi-uav dynamic area coverage algorithm for complex environment," *Neurocomputing*, vol. 595, p. 127904, 2024.
[6] G. B. Wijaya and T. A. Tamba, "Area coverage maximization of multi uavs using multi-agent reinforcement learning," pp. 1–4, 2023.
[7] "https://www.python.org/."