UAV Target Tracking with Bandit-Based Data Fusion

Yang Lv[1,2], Guochao Fan[1], Mengzhen Li[2], Xiongjun Liu[1], Pengqing Liu[2], and Yapu Zhang[2(✉)]

[1] Beijing Jinghang Computation and Communication Research Institute, Beijing 100074, People's Republic of China
fanguochao@yeah.net

[2] Institute of Operations Research and Information Engineering, Beijing University of Technology, Beijing 100124, People's Republic of China
lmz9710@emails.bjut.edu.cn, zhangyapu@bjut.edu.cn

Abstract. Unmanned Aerial Vehicles have emerged as the optimal solution for target tracking due to their low cost and high maneuverability. We study the problem of target tracking in partially observable adversarial environments. Building on previous research, we develop a data fusion algorithm to optimize target tracking. To enable a single UAV to track a hard-to-observe target with limited sensor capabilities, we construct a target environment tracking model along with its corresponding reward function. In the algorithm design, we incorporate the Exp4-IX algorithm from the framework of an adversarial multi-armed bandit with advice, and we prove that the regret bound of this algorithm exhibits sub-linear growth. In numerical experiments, the Exp4-IX algorithm integrates the Previous Position algorithm, Particle Filtering algorithm, and Trajectory Fitting algorithm, and is benchmarked against the Average Fusion algorithm. The results demonstrate its effectiveness in online fusion for smooth trajectory scenarios. This integration allows the UAV to predict the target's position with greater accuracy compared to other algorithms.

Keywords: Unmanned Aerial Vehicles · Target tracking · Multi-armed bandit · Data fusion · Exp4-IX algorithm

1 Introduction

Unmanned Aerial Vehicles (UAVs), characterized by their low cost, small size, and high maneuverability, possess significant application potential. UAVs can be equipped with IoT devices, sensors, and processors, enabling them to perform various functions, including communication, flight, and computation. UAVs are widely employed in various tasks, such as maritime operations [13], disaster relief [14], communication coverage [9], navigation [17], and target tracking [11,12]. Among these, target tracking is one of the most notable applications, as UAVs can cover vast areas and access locations difficult for humans to reach.

D. Du et al. (Eds.): COCOA 2024, LNCS 15434, pp. 278–286, 2025.
https://doi.org/10.1007/978-981-96-4445-2_23

There is extensive research on tracking a single dynamic target with a single UAV, including the Actor-Critic reinforcement learning algorithm [2], YOLO algorithm [5], and DQN algorithm [1]. In the study by [5], the YOLO algorithm is used for real-time target detection when the UAV is stationary, while [1] demonstrates that the DQN algorithm performs better when the line of sight is obstructed by obstacles. When this data is transmitted to the fusion center, it is essential to evaluate each scheme and select the optimal advice to ensure that the target remains within the UAV's surveillance range (Fig. 1).

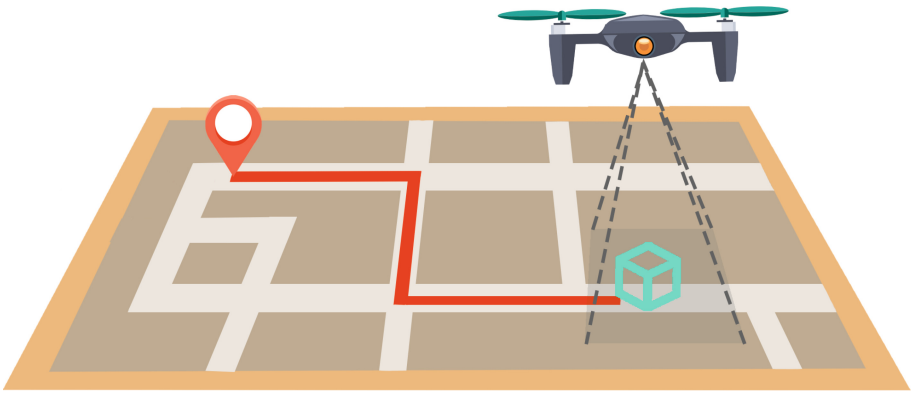

Fig. 1. A single dynamic target with a single UAV in adversarial environments: In target tracking, partial observability refers to the fact that the UAV can only observe the square-shaped region depicted in the image; if the object moves out of this region, it cannot be detected. An adversarial environment refers to a scenario where the target's movement direction may be completely opposite to the UAV's, requiring the UAV to respond swiftly. Under these challenges, we consider the strategy provided by multiple experts, enabling the UAV to anticipate the object's next position in advance.

In scenarios where multiple algorithms or experts provide advice, the modeling approach for tracking a single dynamic target with a single UAV is as follows. The UAV has k possible actions, set as four actions in this paper: forward, backward, left, and right. Let x_t denote the action reward of the UAV at each time step, which will be described in detail later. Our goal is to identify the optimal expert by solving the following optimization problem:

$$\min R_n = \min \mathbb{E}\left[\max_{m\in[M]} \sum_{t=1}^{n} E_m^{(t)} x_t - \sum_{t=1}^{n} X_t\right] \tag{1}$$

where n represents the total number of time steps during the UAV's flight, R_n is the expected difference between the optimal expert's reward and the actual action reward, $E_m^{(t)}$ is the probability provided by the m-th expert for each action at time t, and X_t is the reward after the actual action is taken. If the best expert is found from the beginning, then the regret value is 0.

$\hat{R}_n$ is defined as the regret without expectation, that is, $\mathbb{E}[\hat{R}_n] = R_n$. We demonstrate that for every $\delta \in (0,1)$, there exists an algorithm such that, with probability at least $1-\delta$, the regret $\hat{R}_n$ is upper-bounded.

Tracking a target using a UAV is a dynamic process that involves two key settings: partial observability and an adversarial environment.

a) **Partial Observability:** Although the UAV is equipped with wide-field and long-range radar sensors, its coverage is limited to a rectangular area 60 km ahead in the direction of the aircraft's speed, with a width of 80 km. When the target may not be visible, we use online learning methods to evaluate the effectiveness of UAV actions to assess the accuracy of expert advice and adjust in real-time based on the results. This process requires a proper balance between exploration and exploitation, commonly referred to as the bandit problem. If poor expert advice is selected, the UAV will lose track of the target, reducing the action reward to zero and rapidly increasing the regret.
b) **Adversarial Environment:** The target chooses to move in the opposite direction of the UAV's flight path. The target's movement is unpredictable and may even be adversarial [4]. The experts must not only accurately predict the state of the target but also respond to the target's adversarial behavior.

In the bandit problem of UAV tracking, some studies [15] have explored the adversarial bandit problem, but using the Fixed-Share algorithm [3] increases the overall upper bound of the algorithm's regret. Other studies [6,8,16] assume that the environment follows a randomly independent model; however, in reality, the target's flight trajectory is often adversarial.

Contribution: Our algorithm employs the bandits with expert advice algorithm for single UAV and single target tracking, which integrates multiple algorithms and enhances overall performance. Based on Gergely Neu's Exp4-IX (Implicit eXploration) algorithm [10], we prove that when $\delta < O(1/M)$, our algorithm has a superior high-probability upper bound ([10] Theorem 1). The condition $\delta < O(1/M)$ is relatively easy to satisfy because δ is particularly small.

2 Bandit with Expert Advice

We present the background of the "Bandits with Expert Advice" problem. There are n rounds of interaction between the UAV and the target. In the t-th round, the UAV can choose from k actions, where the directions of movement are restricted to forward, right, downward, and left. In each round, M experts provide flight advice to the UAV based on the previous conditions. Let the advice of the M experts be $E_1^{(t)}, E_2^{(t)}, ..., E_M^{(t)} \in [0,1]^k$, where $\sum_{i=1}^{k} E_{m,i}^{(t)} = 1$ for all $m \in \{1, 2, ..., M\}$. The reward $x_t \in [0,1]^k$ represents the reward the UAV receives for each action, typically assumed to be between 0 and 1.

Since the UAV's observations are partially observable, suppose the UAV's current position is denoted as (x, y), and it can only detect the area within a

range x_s in front of it and a range y_s on both sides. Therefore, the detection range forms a rectangular area. We define the UAV's detection range as:

$$\begin{cases} [x - y_s, x + y_s] \times [y, y + x_s] & \text{if the UAV is moving forward;} \\ [x, x + x_s] \times [y - y_s, y + y_s] & \text{if the UAV is moving right;} \\ [x - y_s, x + y_s] \times [y - x_s, y] & \text{if the UAV is moving downward;} \\ [x - x_s, x] \times [y - y_s, y + y_s] & \text{if the UAV is moving left.} \end{cases} \tag{2}$$

The reward is not simply set based on whether the UAV observes the target (rewarding 0 when observed and 1 when not observed), but rather it is determined by the distance between the UAV and the target. We assume that the closer the distance, the higher the reward, as the target is more likely to appear within the UAV's field of view. Let d_t^i represent the distance after the UAV executes the i-th action, then we define the reward $x_t^{(i)}$ for taking the i-th action in the t-th round as:

$$x_t^{(i)} = \begin{cases} \min\{\frac{1}{d_t^i}, 1\} & \text{if the target is within the field of view;} \\ 0 & \text{if the target is outside the field of view.} \end{cases} \tag{3}$$

This setup ensures that the target is within the UAV's detection range and encourages the UAV to get as close to the target as possible.

In the adversarial environment, our algorithm guarantees a sub-optimal expected upper bound, with the regret value exhibiting sub-linear growth as the number of rounds increases. This indicates that the UAV can not only approach the target but also continuously keep it in sight.

3 Exp4-IX And Regret Upper Bound

Exp4-IX is a standard algorithm for adversarial bandit with advice. It is named Exp4-IX because its full name is the "Exponential-weight algorithm for Exploration and Exploitation with Expert advice - Implicit eXploration". The Exp4-IX algorithm sets the initial weights of the experts as $Q_t = (1/M, 1/M, \ldots, 1/M) \in [0, 1]^M$ and executes the following two steps at each time step:

1. Before the UAV executes, obtain the advice from the M experts and use a linear transformation to determine the UAV's flight strategies.
2. After the UAV executes, receive a reward based on the flight state and update the experts' weights.

The detailed algorithm flow is presented in Algorithm 1. To simplify the proof, the reward values x_{ti} are replaced by the loss values $y_{ti} = 1 - x_{ti}$.

It is worth noting that in the algorithm, $\hat{Y}_{ti} = \mathbb{I}\{A_t = i\} y_{ti}/(P_{ti} + \gamma)$, when $\gamma = 0$, the closer P_{ti} gets to 0, the larger the overall variance, leading to very unstable $\tilde{S}_{t,i}$. Therefore, Implicit eXploration is introduced to move Q_t towards a

Algorithm 1: Exp4-IX Algorithm

Input: Number of rounds n, number of actions k, number of experts M, expert advice at different times $E^{(t)} \in [0,1]^{k\times M}$, algorithm parameters η, γ

Initialize $Q_1 = (1/M, \ldots, 1/M)^T \in [0,1]^{M\times 1}$, $S_0 = (0, \ldots, 0)^T_{M\times 1}$ **for** $t = 1, \ldots, n$ **do**

`// Receive advice from` M `experts and give flight suggestions to the UAV`

Receive advice $E^{(t)}$

Choose the action $A_t \sim P_t$, where $P_t = E^{(t)}Q_t$

`// Receive reward based on UAV flight state and update expert weights`

Receive the reward $Y_t = 1 - x_{tA_t}$

Estimate the action rewards: $\hat{Y}_{ti} = \frac{\mathbb{I}\{A_t=i\}Y_t}{P_{ti}+\gamma}$

Propagate the rewards to the experts: $\tilde{Y}_t = E^{(t)T}\hat{Y}_t$

Compute experts' cumulative rewards:

$$\tilde{S}_{t,m} = \tilde{S}_{t-1,m} + \tilde{Y}_{tm} \quad \text{for all } m \in [M]$$

Update the distribution Q_t using exponential weighting:

$$Q_{t+1,m} = \frac{\exp\left(-\eta\tilde{S}_{tm}\right)}{\sum_j \exp\left(-\eta\tilde{S}_{tj}\right)} \quad \text{for all } m \in [M]$$

uniform distribution, increasing the randomness of the actions, which can be seen as a form of "forced exploration" [7]. This increases the probability of selecting other actions. After incorporating γ, the upper regret bound of the algorithm becomes at least a high probability bound of $1 - \delta$.

The first term of Theorem 1 is constant. We provide an upper bound for the second term $2\sqrt{nk\left(\log M + \log(k+1) - \log\delta\right)}$. Compared to the second term in [10] Theorem 2, $\sqrt{(2nk)/(\log M)}\log(2/\delta)$, our upper bound is smaller when $\delta > O(1/(nk))$. Note that $O(1/(nk))$ is a rather trivial upper bound, so the bound presented in this paper is tighter. All omitted proofs are deferred to the journal version of the paper.

Theorem 1. *For a fixed* $\delta \in (0,1)$*, define* $\eta = \sqrt{(\log M + \log(k+1) - \log\delta)/(nk)}$ *,* $\gamma = \eta/2$*. The following holds with probability at least* $1 - \delta$*:*

$$\hat{R}_n \le k\log\frac{k+1}{\delta} + 2\sqrt{nk\left(\log M + \log(k+1) - \log\delta\right)} \tag{4}$$

4 Numerical Evaluation for Exp4-IX

In this experiment, we apply the Exp4-IX algorithm to a scenario involving a single dynamic target with a single UAV. We consider different target trajectories and expert strategies for UAV flight. The experiment examines smooth and adversarial trajectories based on the target's motion, while expert advice includes the previous position algorithm, particle filtering algorithm, trajectory fitting algorithm, and the Exp4-IX algorithm. We evaluate the performance of these expert recommendations, along with the Exp4-IX algorithm, under different conditions in UAV target tracking.

The experimental scenario is set within a square area with a side length of 100 km. The time step is 1 s, and the total simulation time is 2000 s seconds. The UAV's movement direction is restricted to four cardinal directions, with a flight speed of 280 m per second. The detection range is a rectangular area extending 10 km forward and 8 km to each side. The target's speed is controlled between 100 and 280 m per second, and the scenario is set with smooth trajectories.

Smooth trajectory generation: Twenty points are selected on a plane, and a smooth curve is fitted through these points. The fitted curve is then further subdivided to ensure that the distance between any two points does not exceed the target's speed. The number of subdivided points does not exceed 2000; any excess points are truncated.

Expert recommendations are categorized as follows:

- **Previous Position Algorithm:** If the UAV detects the target's position in the previous time step, it calculates the probability of moving in each direction based on the spatial relationship between the UAV and the target. If the target was not detected in the previous step, equal probabilities are assigned to all directions to increase the chance of detecting the target.
- **Particle Filtering Algorithm:** A set of particles representing the target's state is generated at the start of the filter. Based on a model, random noise is added to the particles to predict the target's position at the next time step. The UAV's next movement direction is then determined by comparing the predicted position with the UAV's position. The particle weights are updated based on the UAV's observations, and resampling is performed as needed to avoid particle degradation.
- **Trajectory Fitting Algorithm:** The UAV records data based on observed target trajectories at each step. It then predicts the target's trajectory position at each step to determine the next flight direction.
- **Average Fusion Algorithm:** This online algorithm integrates the three aforementioned algorithms by averaging the probabilities provided by the first three algorithms.
- **Exp4-IX Algorithm:** This online algorithm integrates the three aforementioned algorithms, dynamically adjusting their weights to select the optimal strategy (Fig. 2).

The code has been open-sourced on GitHub: https://github.com/lvymath1/UAV_tracking_bandit_algorithm.

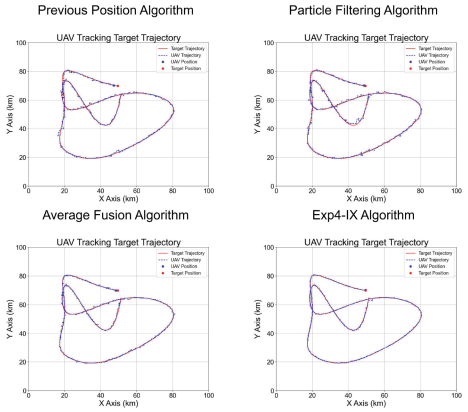

Fig. 2. Effect of the Previous Position Algorithm, Particle Filtering Algorithm, Average Fusion Algorithm, and Exp4-IX Algorithm on UAV target tracking.

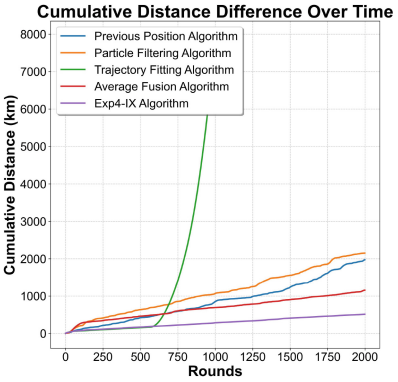

Fig. 3. Image showing the cumulative distance of five algorithms in UAV target tracking.

All simulations were conducted on a Windows laptop equipped with an Intel Core i7-12700H CPU @ 2.30 GHz and 32 GB of RAM, running Python 3.9.

In smooth environments, the target's direction of movement does not change due to the UAV's position. The target's flight path is smooth, with minimal reward fluctuations, making it less adversarial. The Exp4-IX algorithm outperforms other algorithms (see Fig. 3), followed by the Average Fusion algorithm. This indicates that under smooth conditions, the fusion algorithm is superior to individual algorithms, and the Exp4-IX algorithm is a more optimal fusion algorithm. Among the other three algorithms, the Previous Position algorithm performs the best, followed by the Particle Filtering algorithm, but it exhibits greater trajectory volatility compared to Exp4-IX. In the Trajectory Fitting algorithm, the UAV loses track of the target during the tracking process, causing it to predict the trajectory-based only on previously detected target positions, thus gradually deviating from the target.

5 Conclusion

We study the problem of target tracking using UAVs in partially observable adversarial environments and introduce the Exp4-IX algorithm for online data fusion. This algorithm not only has theoretical guarantees but also performs exceptionally well in numerical experiments. Our study demonstrates that Exp4-IX can effectively integrate multiple UAV algorithms, enhancing the UAVs' target-tracking capabilities. However, the algorithm has the following limitations: 1) If the UAV's detection range is not a rectangular area but another

shape, the reward function needs to be adjusted. 2) Theoretically, the upper bound of the UAV's regret may be lower than the best expert's error, so if the optimal expert strategy is sufficiently good, Exp4-IX may not perform better than the optimal expert strategy. In future work, we plan to extend the strategy from a single UAV to a collective strategy involving multiple UAVs to track more targets under conditions of partial communication.

References

1. Bhagat, S., Sujit, P.B.: UAV target tracking in urban environments using deep reinforcement learning. In: International conference on unmanned aircraft systems, pp. 694–701. IEEE, Athens (2020)
2. Elhussein, A., Miah, M.S.: A novel model-free actor-critic reinforcement learning approach for dynamic target tracking. In: IEEE Midwest Industry Conference, pp. 1–6. IEEE, Champaign (2020)
3. Herbster, M., Warmuth, M.K.: Tracking the best expert. Mach. Learn. **32**(2), 151–178 (1998)
4. Jin, C., Jin, T., Luo, H., Sra, S., Yu, T.: Learning adversarial Markov decision processes with bandit feedback and unknown transition. In: International Conference on Machine Learning, pp. 4860–4869. PMLR (2020)
5. Lai, Y.C., Huang, Z.Y.: Detection of a moving UAV based on deep learning-based distance estimation. Remote Sens. **12**(18), 1–28 (2020)
6. Landgren, P., Srivastava, V., Leonard, N.E.: Distributed cooperative decision making in multi-agent multi-armed bandits. Automatica **125**(109445), 1–13 (2021)
7. Lattimore, T., Szepesvári, C.: Bandit Algorithms. Cambridge University Press (2020)
8. Lee, K.M.B., et al.: An upper confidence bound for simultaneous exploration and exploitation in heterogeneous multi-robot systems. In: International Conference on Robotics and Automation, pp. 8685–8691. IEEE (2021)
9. Lv, Y., Wu, C., Xu, D., Yang, R.: H-hop independently submodular maximization problem with curvature. High-Confidence Comput. **4**(100208), 1–7 (2024)
10. Neu, G.: Explore no more: improved high-probability regret bounds for non-stochastic bandits. In: Advances in Neural Information Processing Systems, pp. 1–9. NIPS, Montréal (2015)
11. Ojha, S., Sakhare, S.: Image processing techniques for object tracking in video surveillance-a survey. In: International Conference on Pervasive Computing, pp. 1–6. IEEE, St. Louis (2015)
12. Sun, S., Liu, Y., Guo, S., Li, G., Yuan, X.: Observation-driven multiple UAV coordinated standoff target tracking based on model predictive control. Tsinghua Sci. Tech. **27**(6), 948–963 (2022)
13. Wang, G., Wang, F., Wang, J., Li, M., Gai, L., Xu, D.: Collaborative target assignment problem for large-scale UAV swarm based on two-stage greedy auction algorithm. Aerosp. Sci. Tech. **149**(109146), 1–11 (2024)
14. Xu, W., et al.: Approximation algorithms for the generalized team orienteering problem and its applications. IEEE/ACM Trans. Netw. **29**(1), 176–189 (2020)
15. Xu, Z., Lin, X., Tzoumas, V.: Bandit submodular maximization for multi-robot coordination in unpredictable and partially observable environments. arXiv preprint arXiv:2305.12795, pp. 1–14 (2023)

16. Zhang, C., Hoi, S.C.: Partially observable multi-sensor sequential change detection: a combinatorial multi-armed bandit approach. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 5733–5740 (2019)
17. Zhu, K., Zhang, T.: Deep reinforcement learning based mobile robot navigation: a review. Tsinghua Sci. Tech. **26**(5), 674–691 (2021)