AntSearcher: An Ant Colony Optimization-Based Framework for Moving Target Search with UAVs

Qingxia Li, Ani Dong, Yuhui Zhang*, and Wenhong Wei

***Abstract*—The Moving Target Search (MTS) problem presents a complex optimization challenge in dynamic environments. This paper introduces AntSearcher, an Ant Colony System (ACS) algorithm tailored to efficiently address the MTS problem. AntSearcher utilizes a custom solution construction graph, where ants traverse edges to identify UAV paths maximizing the probability of detecting moving targets. The algorithm incorporates problem-specific heuristic and pheromone tensors to steer the search process. It includes a time-varying telescope heuristic, which exploits the target's initial belief map to enhance the early search phase, and a reinitialization mechanism to mitigate premature convergence. Extensive experiments across diverse search scenarios demonstrate that AntSearcher surpasses state-of-the-art evolutionary optimizers, showcasing its superior effectiveness and robustness.**

***Index Terms*—Ant Colony Optimization, UAV, Moving Target Search, Swarm Intelligence.**

I. Introduction

Unmanned aerial vehicles (UAVs) have garnered significant research interest and are increasingly deployed in a wide array of practical applications, particularly in surveillance and rescue operations [1]–[3]. Their ability to operate in harsh and inaccessible environments, equipped with advanced sensors, makes them invaluable for tasks requiring rapid response and data collection. In search and rescue (SAR) scenarios, UAVs significantly enhance disaster response capabilities by providing real-time aerial imagery, thermal imaging for locating survivors, and AI-based damage analysis and coordination.

In searching for lost or moving targets using UAVs, a critical challenge arises in the form of the moving target search (MTS) problem, where the objective is to determine optimal flight paths that maximize the probability of detecting dynamic targets within a constrained time frame, often referred to as the “golden time” during which detection likelihood is highest [4]. This problem is inherently complex due to uncertainties in target motion, environmental factors such as terrain and weather, imperfect sensor models, and the exponential growth of possible search paths, rendering it NP-hard in high-dimensional scenarios [5], [6]. MTS is typically formulated within a probabilistic framework, incorporating Bayesian updates to a belief map representing target location probabilities over time, and has applications in search and rescue, surveillance, and maritime operations where targets may evade or move unpredictably.

Existing approaches to the MTS problem have employed a variety of optimization techniques to address its inherent complexity. Early methods, such as greedy search with one-step or k-step look-ahead strategies, focus on immediate or short-term path planning but often fail to optimize long-term detection probabilities due to their myopic nature [7], [8]. More advanced metaheuristic algorithms, including ant colony optimization (ACO) [9]–[12], genetic algorithms (GA) [13]–[16], differential evolution (DE) [17]–[20], and particle swarm optimization (PSO) [21]–[24] have been adapted to handle dynamic target scenarios by leveraging population-based searches, though they may struggle with computational efficiency or premature convergence. The classical PSO, as the original version of the algorithm, serves as a foundational method where particles adjust their positions based on personal and global bests [25], while variants like the motion-encoded particle swarm optimization (MPSO) [4] build upon this by encoding search trajectories as particle positions. Other techniques, including Bayesian optimization approaches (BOA) [26], cross-entropy optimization (CEO) [27], branch and bound [28], limited depth search [29], and gradient descent methods [30], [31], incorporate probabilistic models and multi-agent systems to varying degrees, yet their effectiveness often depends on specific problem constraints and real-time requirements, necessitating further refinement for robust UAV-based MTS solutions. Efficient solutions remain essential for real-time deployment with single or multi-UAV systems, particularly in time-sensitive rescue operations where rapid adaptability and computational efficiency are critical.

The ACO algorithm, inspired by the foraging behavior of ants, is a metaheuristic approach that simulates the pheromone-based communication used by ants to find the shortest paths to food sources. In ACO, a colony of artificial ants constructs solutions by moving through a graph, depositing pheromone trails on edges based on solution quality, which guides subsequent ants toward promising areas. Through iterative reinforcement of successful paths and evaporation of weaker trails, ACO effectively balances exploration and exploitation, making it well-suited for solving complex combinatorial optimization problems [9], [32]. The Ant System (AS) [33], introduced by Marco Dorigo, serves as the foundational model, initially applied to the Traveling Salesman Problem (TSP) to establish the core pheromone-based mechanism. The Ant Colony System (ACS) [34] enhances AS by incorporating local pheromone updates and a candidate list strategy to improve solution quality and convergence speed. Additionally, the Max-Min Ant System (MMAS) [35] constrains pheromone values within fixed bounds to prevent premature convergence, offering robustness across various optimization tasks. These variants have

Q. Li and A. Dong are with the School of Artificial Intelligence, Dongguan City University, China.

Y. Zhang and W. Wei are with the School of Computer Science and Technology, Dongguan University of Technology, China.

*Corresponding author:yhzhang@dgut.edu.cn

laid a solid groundwork for ACO's evolution, demonstrating its versatility in tackling diverse problems. Given its ability to adaptively search dynamic environments and optimize path-based solutions, ACO emerges as a promising candidate for addressing the MTS problem, where it can potentially enhance UAV path planning by leveraging pheromone-guided heuristics to maximize detection probabilities in real-time scenarios.

This paper presents AntSearcher, a novel ACO approach tailored to address the MTS problem. AntSearcher features a carefully designed solution construction graph that models the search space, enabling ants to explore UAV paths efficiently. Corresponding heuristic and pheromone tensors are integrated to steer the ants' solution construction process. The algorithm incorporates a problem-specific telescope heuristic to guide the initial search phase using the target's initial belief map, complemented by a time-varying heuristic update that adapts to evolving target dynamics. Furthermore, AntSearcher includes a stagnant detection mechanism with pheromone matrix reinitialization to prevent premature convergence, alongside a dynamic setting for the initial pheromone value τ_0 to enhance search flexibility. These advancements collectively enhance UAV path optimization, improving the probability of detecting moving targets in complex scenarios.

The remainder of this paper is structured as follows. Section II outlines the problem formulation for the MTS problem. Section III provides a detailed description of the proposed AntSearcher algorithm. Section IV presents experimental evaluations to assess the algorithm's performance. Finally, Section V offers conclusions and highlights potential directions for future research.

II. Problem Formulation

The MTS problem is a critical challenge in UAV operations, particularly in applications such as search and rescue, where the objective is to locate a dynamically moving target within a constrained time window. This problem is inherently complex due to uncertainties in target motion, environmental conditions, and sensor limitations, necessitating a probabilistic framework to model the search process effectively. In this study, we adopte the formulation presented in [4] and provide more precise mathematical details.

A. Target Model

The target is represented by its location $\mathbf{x}_t \in \mathbb{R}^2$ at time t, where $\mathbf{x}_t = [x_t, y_t]$ denotes the 2D coordinates within the search space $\mathcal{S}$, discretized into a grid of $S_r \times S_c$ cells. Initially, the target's location is uncertain, modeled by a probability density function (PDF) $b(\mathbf{x}_0)$, typically a multivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ centered at the last known position $\boldsymbol{\mu}_0$ with covariance $\boldsymbol{\Sigma}_0$, normalized such that $\sum_{\mathbf{x}_0 \in \mathcal{S}} b(\mathbf{x}_0) = 1$ assuming the target resides within $\mathcal{S}$. The target's motion is governed by a stochastic process, approximated as a first-order Markov model with a transition probability $p(\mathbf{x}_t|\mathbf{x}_{t-1})$, which defines the likelihood of the target moving from cell $\mathbf{x}_{t-1}$ to $\mathbf{x}_t$ based on predefined motion patterns (e.g., random walk or directed drift). For simplicity, we assume a conditionally deterministic case where $p(\mathbf{x}_t|\mathbf{x}_{t-1})$ depends solely on $\mathbf{x}_0$, enabling prediction of the target's path given its initial state.

B. Sensor Model

The UAV is equipped with a sensor that generates an observation z_t at each time step t, modeled as a binary random variable: $z_t \in \{D_t, \overline{D_t}\}$, where D_t indicates target detection and $\overline{D_t}$ indicates non-detection. The sensor's observation likelihood is defined by $p(z_t|\mathbf{x}_t)$, which reflects the probability of detecting the target given its true location. This likelihood is parameterized by a detection probability function, often modeled as a Gaussian $p(D_t|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_t; \mathbf{x}_s(t), \sigma_s^2 \mathbf{I})$, where $\mathbf{x}_s(t)$ is the sensor's position at time t and σ_s^2 is the sensor's detection variance. The non-detection probability is then $p(\overline{D_t}|\mathbf{x}_t) = 1 - p(D_t|\mathbf{x}_t)$, accounting for sensor imperfections such as noise or occlusion.

C. Belief Map Update

The belief map $b(\mathbf{x}_t)$ represents the posterior probability of the target being at $\mathbf{x}_t$ given all observations up to time t, denoted as $z_{1:t} = \{z_1, \ldots, z_t\}$. This map is updated recursively using a Bayesian approach. The prediction step propagates the previous belief $b(\mathbf{x}_{t-1}) = p(\mathbf{x}_{t-1}|z_{1:t-1})$ forward via the Chapman-Kolmogorov equation:

$$\hat{b}(\mathbf{x}_t) = \sum_{\mathbf{x}_{t-1} \in \mathcal{S}} p(\mathbf{x}_t|\mathbf{x}_{t-1}) b(\mathbf{x}_{t-1}), \tag{1}$$

where $\hat{b}(\mathbf{x}_t)$ is the predicted belief. The update step incorporates the current observation z_t using Bayes' rule:

$$b(\mathbf{x}_t) = \eta_t p(z_t|\mathbf{x}_t) \hat{b}(\mathbf{x}_t), \tag{2}$$

where $\eta_t = \left(\sum_{\mathbf{x}_t \in \mathcal{S}} p(z_t|\mathbf{x}_t) \hat{b}(\mathbf{x}_t)\right)^{-1}$ is the normalization constant ensuring $\sum_{\mathbf{x}_t \in \mathcal{S}} b(\mathbf{x}_t) = 1$. This process refines the belief map dynamically as new sensor data is acquired.

D. Objective function

The goal of the MTS problem is to determine an optimal UAV path $\mathbf{O} = \{\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_N\}$, where $\mathbf{o}_t \in \mathcal{S}$ is the UAV's position at time t over a finite horizon N, that maximizes the cumulative probability of detecting the target. The probability of non-detection at time t is $r_t = \sum_{\mathbf{x}_t \in \mathcal{S}} p(\overline{D_t}|\mathbf{x}_t) \hat{b}(\mathbf{x}_t)$, and the joint probability of non-detection up to time t is $R_t = \prod_{k=1}^{t} r_k$. The probability of first detection at time t is $p_t = R_{t-1}(1 - r_t)$, with $R_0 = 1$. The cumulative detection probability over N steps is thus:

$$P_N = \sum_{t=1}^{N} p_t = 1 - R_N, \tag{3}$$

and the objective function to maximize is:

$$J(\mathbf{O}) = P_N, \tag{4}$$

subject to constraints on UAV flight time, energy, and kinematic limits. This formulation seeks to optimize $\mathbf{O}$ to balance exploration and exploitation, ensuring high detection likelihood within the operational constraints.

III. AntSearcher

Inspired by the robust search capabilities of ACO and the need to tackle the inherent complexity of the MTS problem, we have developed AntSearcher, an ACO-based approach tailored for this challenge. We first outline the overall framework of AntSearcher, providing a clear procedural foundation. Next, we introduce the solution construction graph, a structured representation through which artificial ants navigate to generate candidate solutions. We then propose a problem-specific heuristic, termed the "Telescope heuristic," which leverages the target's initial belief map to effectively guide the ants' search process. Subsequently, we detail the solution construction steps undertaken by the artificial ants, alongside the corresponding pheromone update procedures, ensuring a comprehensive understanding of the optimization dynamics. Finally, a stagnant detection and reinitialization mechanism is integrated to prevent the ants from becoming trapped in local optima, enhancing the algorithm's global search capability.

A. Overall Framework of AntSearcher

AntSearcher inherits the basic procedures of ACS [34]. To provide an overview of the proposed algorithm, Fig. 1 presents its flowchart. The algorithm begins by constructing the solution graph based on the problem instance, followed by initialization of the corresponding heuristic and pheromone tensors. It then enters an iterative loop in which a colony of ants incrementally builds candidate solutions by traversing edges of the construction graph. Each ant constructs its tour according to the state-transition rule of ACS, and a local pheromone update is performed after every move of every ant.

To account for the continuously moving target and the consequent decay in reliability of the initial belief map $b(\mathbf{x}_0)$, the heuristic tensor is dynamically reduced during the search. Once all ants have completed their tours, the quality of each tour is evaluated. Only the ant that found the best tour in the current iteration (the iteration-best ant) is allowed to perform a global pheromone update. Additionally, the initial pheromone value τ_0 is periodically adjusted based on the best-so-far tour.

The algorithm then checks whether the termination criterion is satisfied. If so, it outputs the best tour found throughout the entire run. Otherwise, it monitors stagnation by counting the number of consecutive iterations during which the globally best solution has not improved. When stagnation is detected, the pheromone tensor is reinitialized using the current value of τ_0, thereby encouraging the ants to restart exploration and escape local optima.

B. Solution Construction Graph

The search space $\mathcal{S}$ is discretized into a grid of $S_r \times S_c$ cells, where each cell corresponds to a node in the solution construction graph. All ants start from the same initial position $\mathbf{o}_0$. When an ant is located at cell $c_{i,j}$, it can move to any of the eight adjacent cells: $c_{i-1,j-1}$, $c_{i-1,j}$, $c_{i-1,j+1}$, $c_{i,j-1}$, $c_{i,j+1}$, $c_{i+1,j-1}$, $c_{i+1,j}$, and $c_{i+1,j+1}$, as illustrated in Fig. 2(a). Consequently, in the solution construction graph, the node representing cell $c_{i,j}$ has eight outgoing edges, one for each possible move.

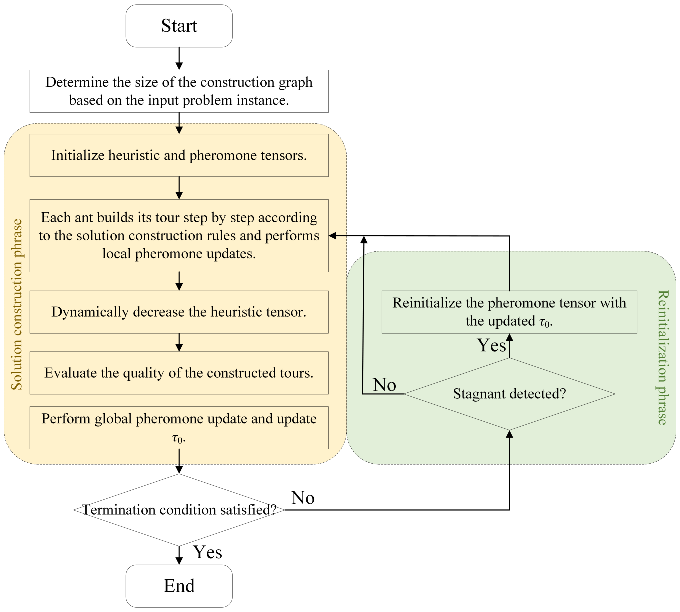

Start
Determine the size of the construction graph based on the input problem instance.
Solution construction phrase
Initialize heuristic and pheromone tensors.
Each ant builds its tour step by step according to the solution construction rules and performs local pheromone updates.
Dynamically decrease the heuristic tensor.
Evaluate the quality of the constructed tours.
Perform global pheromone update and update τ_0.
Reinitialization phrase
Reinitialize the pheromone tensor with the updated τ_0.
Yes
No
Stagnant detected?
Termination condition satisfied?
No
Yes
End

Fig. 1. Flowchart of AntSearcher.

Figure 2(b) illustrates the complete solution construction graph for the moving-target search (MTS) problem. Because eight actions are possible from each cell, both pheromone values and heuristic information are stored in third-order tensors of size $S_r \times S_c \times 8$, rather than the two-dimensional matrices employed in conventional ant colony optimization algorithms (see Fig. 2(c) for a visual representation of the tensor structure). The third dimension of these tensors indexes the eight possible movements from any given cell. For notational convenience, the eight possible movements are encoded with integers 0 to 7, corresponding to the directions North, North-East, East, South-East, South, South-West, West, and North-West, respectively.

C. Heuristic Design

1) The "Telescope" Heuristic: The heuristic information is a critical component of ant colony optimization algorithms, as it biases the search toward promising regions of the solution space. In AntSearcher, we propose a problem-specific heuristic tensor, termed the Telescope heuristic, that exploits prior knowledge encoded in the initial belief map $b(\mathbf{x}_0)$, which represents the probability distribution of the target's location at time zero. This heuristic is specifically designed to guide ants toward constructing high-quality initial solutions.

The heuristic value $\eta(i,j,k)$ associated with moving from cell $c_{i,j}$ in direction $k \in \{0,1,\ldots,7\}$ is computed by summing the positive belief increments encountered along the ray in direction k until a decrease occurs. Formally,

$$\begin{aligned}\eta(i,j,k) = \sum_{d=1}^{n} &[b(i+\delta(k,0)\cdot d,\ j+\delta(k,1)\cdot d) \\ &-b(i+\delta(k,0)\cdot(d-1),\ j+\delta(k,1)\cdot(d-1))]^{+},\end{aligned} \tag{5}$$

where $\delta(k,0)$ and $\delta(k,1)$ are the row and column offsets for direction k (e.g., North-West corresponds to $\delta(7) =$

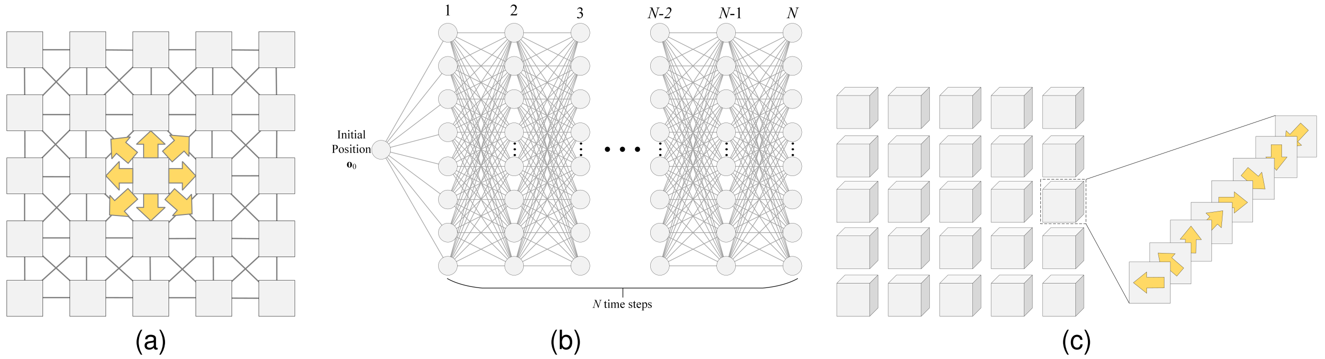


Fig. 2. Graph representation of the MTS problem. (a) Eight possible movements from a given cell $c_{i,j}$ on the discretized grid. (b) Solution Construction Graph. (c) Structure of the third-order pheromone and heuristic tensors (size $S_r \times S_c \times 8$).

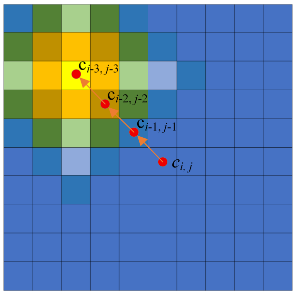


Fig. 3. Example computation of the Telescope heuristic value $\eta(i,j,7)$ (North-West direction) from the initial belief map $b(\mathbf{x}_0)$. The summation follows the ascending belief gradient until the first decrease.

$(-1,-1)$), and the summation stops at the largest n such that the belief value is still strictly increasing along the ray (i.e., the term inside the brackets remains positive). The operator $[x]^+ = \max(x,0)$ is implicitly applied by terminating the sum when a decrease is observed.

Figure 3 illustrates the computation of $\eta(i,j,7)$ (North-West direction). In this example, $c_{i-3,j-3}$ is the farthest cell along the ray where the belief continues to increase, yielding

$$\begin{aligned}\eta(i,j,7) = [b(i-1,j-1) - b(i,j)] + [b(i-2,j-2) \\ -b(i-1,j-1)] + [b(i-3,j-3) - b(i-2,j-2)].\end{aligned} \tag{6}$$

Intuitively, the Telescope heuristic rewards directions that immediately lead toward regions of increasing target probability, thereby encouraging ants to move “uphill” on the initial belief landscape at the early stages of the search.

2) Dynamic Decay of Heuristic Influence: Because the target moves over time according to its own dynamics, the reliability of the initial belief map $b(\mathbf{x}_0)$ decays as the search progresses. To reflect this, the influence of the heuristic tensor is progressively reduced with both elapsed time t and the number of algorithm iterations It. At each update, the entire heuristic tensor is scaled as follows:

$$\eta \leftarrow 1 - \exp\Big(-100\big(\tfrac{t}{N} + \tfrac{It}{\text{MaxIt}}\big)\Big) \cdot (1-\eta), \tag{7}$$

where N is the planning horizon and MaxIt is the maximum number of iterations. This formulation ensures a smooth transition from strong heuristic guidance in the early phase to nearly heuristic-free exploration in later stages, allowing the algorithm to rely increasingly on updated pheromone information that reflects the target's actual motion.

D. Solution Construction and Pheromone Update

Before entering the solution-construction phase, the heuristic tensor η and the pheromone tensor τ are initialized. All entries of the pheromone tensor are set to the same initial value $\tau_0 = 0.2/m$, where m is the number of ants in the colony. Every ant starts from the fixed initial position $\mathbf{o}_0$ and iteratively traverses the construction graph to build a complete candidate route for the MTS problem.

1) State transition rule: When an ant located at cell $c_{i,j}$ selects its next action $k \in \{0,1,\ldots,7\}$, it applies the pseudo-random-proportional rule inherited from the ACS:

$$k = \begin{cases} \arg\max_{a\in\{0,\ldots,7\}} \left\{\tau(i,j,a)\cdot[\eta(i,j,a)]^\beta\right\} & \text{if } q \le q_0, \\ K & \text{otherwise,} \end{cases} \tag{8}$$

where

$q \sim U[0,1]$ is a uniformly drawn random variable, $q_0 \in [0,1]$ is a user-defined parameter controlling the exploitation/exploration balance, $\beta > 0$ is the heuristic weighting exponent, and K is sampled according to the probability distribution

$$p(k \mid i,j) = \frac{\tau(i,j,k)\cdot[\eta(i,j,k)]^\beta}{\sum_{a=0}^{7}\tau(i,j,a)\cdot[\eta(i,j,a)]^\beta}. \tag{9}$$

The first case (exploitation) directly selects the currently most attractive action, while the second case (biased exploration) performs a probabilistic selection that still favors higher pheromone-heuristic products. The parameter q_0 thus determines the relative frequency of pure exploitation versus biased exploration.

2) Local pheromone update: Immediately after an ant chooses action k at cell $c_{i,j}$ and moves to the next cell, a local pheromone update is performed on the traversed edge:

$$\tau(i,j,k) \leftarrow (1-\rho)\cdot\tau(i,j,k) + \rho\cdot\tau_0, \tag{10}$$

where $\rho \in (0,1)$ is the local evaporation rate. This online step-wise evaporation dynamically reduces the desirability of recently used edges, preventing early convergence to a narrow subset of paths and enabling later ants within the same iteration to explore alternative routes more effectively.

3) Global pheromone update: In AntSearcher, only the iteration-best ant (the ant that constructed the tour with the highest detection probability in the current iteration) is allowed to deposit additional pheromone. This focused reinforcement, combined with the pseudo-random-proportional rule, makes the search more directed toward promising neighborhoods of the current best solution. After all ants have completed their tours, global updating is applied to every edge (i, j, k) belonging to the iteration-best tour:

$$\tau(i, j, k) \leftarrow (1 - \alpha) \cdot \tau(i, j, k) + \alpha \cdot \Delta\tau(i, j, k), \tag{11}$$

where $\alpha \in (0, 1)$ is the global pheromone decay parameter and

$$\Delta\tau(i, j, k) = \begin{cases} \frac{bestProb_{iter}}{m} & \text{if edge } (i, j, k) \in \text{best tour}, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

Here $bestProb_{iter}$ denotes the detection probability of the iteration-best tour. This rule provides greater pheromone reinforcement to edges that contribute to higher-quality solutions, iteratively biasing the colony toward routes with superior expected performance. Edges not used by the iteration-best ant undergo pure evaporation, gradually decreasing their attractiveness.

E. Restart mechanism

To prevent AntSearcher from becoming trapped in local optima, we incorporate a stagnation-detection and pheromone-reinitialization mechanism that promotes renewed exploration. During execution, the algorithm continuously tracks the number of consecutive iterations in which the globally best tour has not improved. If this stagnation counter exceeds a predefined threshold $Stagnation_{max}$, the entire pheromone tensor is reinitialized to the current value of τ_0. This operation effectively erases the reinforcement of previously discovered routes, forcing the ants to forget suboptimal paths and explore alternative trajectories that may offer higher detection probability for the moving target.

Unlike traditional ant colony optimization algorithms, where the initial pheromone level τ_0 remains constant throughout the run, AntSearcher dynamically adapts τ_0 to reflect the quality of the best-so-far solution. After each iteration, if a new globally best tour is found, τ_0 is updated as follows:

$$\tau_0 = \frac{bestProb}{m}, \tag{13}$$

where $bestProb$ is the detection probability achieved by following the best tour discovered so far, and m is the number of ants in the colony.

This adaptive strategy ensures that pheromone levels remain appropriately scaled relative to the current solution quality: when a superior tour is found, the baseline pheromone intensity is increased, thereby accelerating convergence toward high-quality regions upon the next reinitialization, while still allowing sufficient exploration diversity. By combining stagnation-triggered restarts with a quality-dependent τ_0, AntSearcher achieves a balanced trade-off between exploitation of known good paths and exploration of unexplored areas.

IV. Experimental Verification

In this section, we evaluate the performance of AntSearcher through extensive experiments. We first describe the benchmark test cases, which are designed to cover a diverse range of scenarios: the UAV starts from different initial positions, and the target follows various motion dynamics. We then present the comparative results against several state-of-the-art baselines and provide a detailed analysis of AntSearcher's effectiveness and robustness. Finally, an ablation study is conducted to quantify the contribution of each proposed algorithmic component (the Telescope heuristic, dynamic heuristic decay, adaptive τ_0, and stagnation-triggered restart mechanism) to the overall performance.

A. Experimental setup

1) Test Scenarios: We evaluate AntSearcher using the six benchmark scenarios originally proposed in [4]. All scenarios are defined on a search space of the same dimensions 40×40 but differ in the initial UAV position, the target's motion model, and the initial belief map $b(\mathbf{x}_0)$. These scenarios are designed to test the algorithm under increasing levels of difficulty.

Scenario 1: Two high-probability regions located close to each other. The subtle differences in position and belief values between the regions can easily mislead algorithms into selecting the slightly inferior peak.

Scenario 2: Two well-separated high-probability regions situated on opposite sides of the UAV's starting position. The algorithm must correctly decide which region to visit first while preserving sufficient time to inspect the second, requiring a careful balance between immediate reward and future opportunity.

Scenario 3: A single dominant high-probability region. Because search effort is not split, cumulative detection probability is typically higher, making this a relatively easy case that serves as a sanity check.

Scenario 4: Also features a single high-probability region but with different target motion dynamics and initial belief distribution compared to Scenario 3. The altered dynamics test the algorithm's sensitivity to prediction accuracy and temporal alignment of the search path.

Scenario 5: Multiple moderate-probability regions combined with a highly unpredictable (near random-walk) target motion model. The lack of a clear "best" region and the rapid decorrelation of the belief map stress the exploration capability and the dynamic decay mechanism of the heuristic.

Scenario 6: A narrow, diagonally oriented high-probability corridor that moves over time. Effective search requires precise diagonal movements over extended periods, which many grid-based planners struggle to maintain due to discretization artifacts and premature convergence to axis-aligned patterns.

These six scenarios collectively cover a wide spectrum of practical difficulties encountered in MTS problems, allowing a thorough assessment of AntSearcher's robustness and adaptability.

2) Compared algorithms: To assess the effectiveness of the proposed AntSearcher, we compare it against two state-of-the-art metaheuristic algorithms: Motion-encoded Particle Swarm Optimization (MOPSO) [4] and the modified LSHADE-SPACMA (mLSHADE-SPACMA) [36]. These baselines represent specialized approaches to trajectory optimization in dynamic search problems, allowing us to evaluate AntSearcher's performance in terms of detection probability, computational efficiency, and robustness across the benchmark scenarios.

MOPSO is a particle swarm optimization variant tailored specifically for UAV-based moving-target search [4]. Unlike traditional path encodings that rely on discrete position sequences, MOPSO represents candidate trajectories as sequences of motion vectors (combining magnitude and direction), which naturally enforces smooth and kinematically feasible paths. This vector-based formulation enhances swarm coherence, adaptability to evolving target dynamics, and overall search efficiency, as demonstrated by its superior performance over several competing metaheuristics in prior studies.

mLSHADE-SPACMA is an advanced differential evolution algorithm for large-scale numerical optimization [36]. Building upon the CEC 2017 winner LSHADE-SPACMA [37], it introduces three key enhancements: (i) a precise elimination and generation strategy to bolster local exploitation; (ii) a modified semi-parametric mutation operator with rank-based selective pressure for more directed evolutionary progress; and (iii) an elite-based external archive to preserve population diversity and hasten convergence. Comprehensive evaluations on the CEC 2014 and CEC 2017 benchmark suites, alongside real-world applications such as point cloud registration, confirm mLSHADE-SPACMA's competitive edge over other leading optimizers.

3) Parameter settings: The parameters of the competing algorithms are configured as recommended by their respective authors to ensure optimal performance. The source codes for MOPSO [4] and mLSHADE-SPACMA [36] were kindly provided by the original developers, and we used the exact settings reported in their publications. This guarantees that the baseline algorithms operate under their best-known configurations, lending greater credibility to the comparative results.

For AntSearcher, the key hyperparameters are the colony size m, the exploitation bias parameter q_0, the local and global pheromone evaporation rates ρ and α, and the heuristic weighting exponent β. After preliminary tuning, these parameters are fixed to the following values across all scenarios: $m = 10$, $q_0 = 0.9$, $\rho = \alpha = 0.1$, and $\beta = 2$.

All algorithms are implemented in MATLAB and executed on the same computing platform. To ensure a fair comparison, the termination criterion is unified: each run stops after a maximum of 1,000,000 fitness evaluations (path evaluations). Every experiment is repeated 25 independent times with different random seeds, and statistical significance is assessed using the Wilcoxon rank-sum test at a 95% confidence level. Reported performance metrics include the mean and standard deviation of the best detection probability achieved over the 25 runs.

B. Overall Performance

Table I reports the detection probabilities achieved by AntSearcher, MOPSO, and mLSHADE-SPACMA across the six benchmark scenarios, based on 25 independent runs. For each scenario, we present the best, mean, and standard deviation of the detection probability, along with the p-values from pairwise Wilcoxon rank-sum tests comparing AntSearcher against each baseline (significance level $\alpha = 0.05$). Bold values indicate the best mean performance in each row.

AntSearcher outperforms both competitors in four out of six scenarios (Cases 1-4), delivering statistically significant improvements in all of them ($p \ll 0.05$). The gains are particularly pronounced in Scenario 2 (two distant high-probability regions) and Scenario 4, where AntSearcher improves the mean detection probability by approximately 9% and 7.4% over the best baseline, respectively. In Scenario 3 (single high-probability region), AntSearcher again achieves the highest mean and best values, confirming its ability to concentrate search effort effectively when a clear optimal region exists.

In Scenario 5 (highly unpredictable target motion), AntSearcher and mLSHADE-SPACMA obtain virtually identical best values (0.21845), with AntSearcher's mean only marginally lower (0.21489 vs. 0.21634). The difference is not statistically significant ($p = 0.381$), indicating comparable performance under extreme uncertainty.

Scenario 6 (diagonal moving corridor) is the only case where AntSearcher is slightly outperformed by mLSHADE-SPACMA in terms of mean detection probability (0.15219 vs. 0.15761, $p = 0.031$). This suggests that the continuous parameterization favored by differential evolution variants may have a minor advantage in maintaining precise diagonal trajectories over long horizons. Nevertheless, AntSearcher still matches the best single-run result of MOPSO and remains highly competitive.

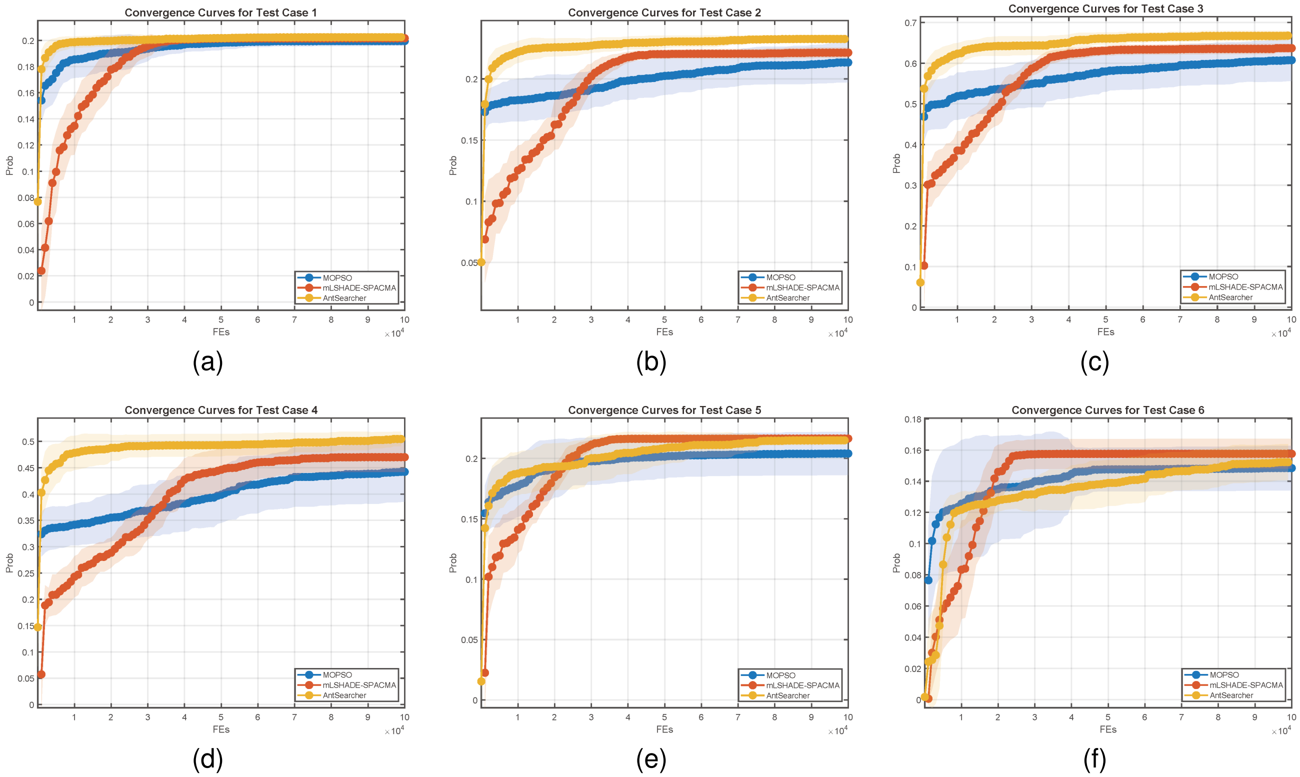
Overall, AntSearcher achieves the highest average rank across all scenarios and demonstrates superior robustness on the majority of tested instances, especially those requiring decisive early commitment to distant high-probability regions (Scenarios 1, 2, and 4). These results validate the effectiveness of the proposed Telescope heuristic, dynamic heuristic decay, adaptive τ_0, and restart mechanism in guiding the colony toward high-quality solutions in dynamic and deceptive search environments.

Figure 4 illustrates the convergence curves of AntSearcher, MOPSO, and mLSHADE-SPACMA across the six benchmark scenarios, showing the evolution of the best detection probability versus the number of fitness evaluations. Figure 5 presents the best trajectories discovered by each algorithm in six test cases.

These visualizations confirm the conclusions drawn from the numerical results: AntSearcher typically exhibits faster initial progress (thanks to the Telescope heuristic) and sustains superior performance throughout the search, especially in Scenarios 1, 2, 3, and 4. In the more challenging Scenarios 5 and 6, AntSearcher still reaches competitive or higher final detection probabilities, while avoiding the premature stagnation occasionally observed in the baselines. The plotted trajectories further reveal that AntSearcher generates smoother,

TABLE I
OVERALL PERFORMANCE COMPARISON (DETECTION PROBABILITY) OF ANTSEARCHER, MOPSO, AND MLSHADE-SPACMA ACROSS THE SIX SCENARIOS (25 INDEPENDENT RUNS). BOLD VALUES INDICATE THE BEST MEAN RESULT IN EACH ROW.

Case	MOPSO					mLSHADE-SPACMA					AntSercher		
	Best	Mean	Std	p-value	Sig.($p < 0.05$)	Best	Mean	Std	p-value	Sig.($p < 0.05$)	Best	Mean	Std
1	0.20269	0.19943	0.00373	1.97E-06	Better	0.20269	0.20162	0.00104	3.18E-04	Better	**0.20372**	**0.20240**	0.00050
2	0.23160	0.21341	0.01626	1.97E-06	Better	0.22890	0.22142	0.00418	5.96E-08	Better	**0.23721**	**0.23263**	0.00315
3	0.67838	0.60748	0.05117	1.97E-06	Better	0.65558	0.63702	0.01092	5.96E-08	Better	**0.68485**	**0.66787**	0.01220
4	0.51191	0.44216	0.05713	1.97E-06	Better	0.50416	0.47012	0.02829	1.01E-05	Better	**0.53737**	**0.50478**	0.01350
5	0.21833	0.20395	0.01791	1.97E-06	Better	**0.21845**	**0.21634**	0.00265	3.81E-01	Not Significant	**0.21845**	0.21489	0.00367
6	0.16256	0.14841	0.01427	1.97E-06	Worse	**0.16259**	**0.15761**	0.00954	3.08E-02	Worse	0.16256	0.15219	0.01149



(a) (b) (c)

(d) (e) (f)

Fig. 4. Convergence curves of AntSearcher, MOPSO, and mLSHADE-SPACMA over the six benchmark scenarios (best detection probability versus number of fitness evaluations, averaged over 25 independent runs).

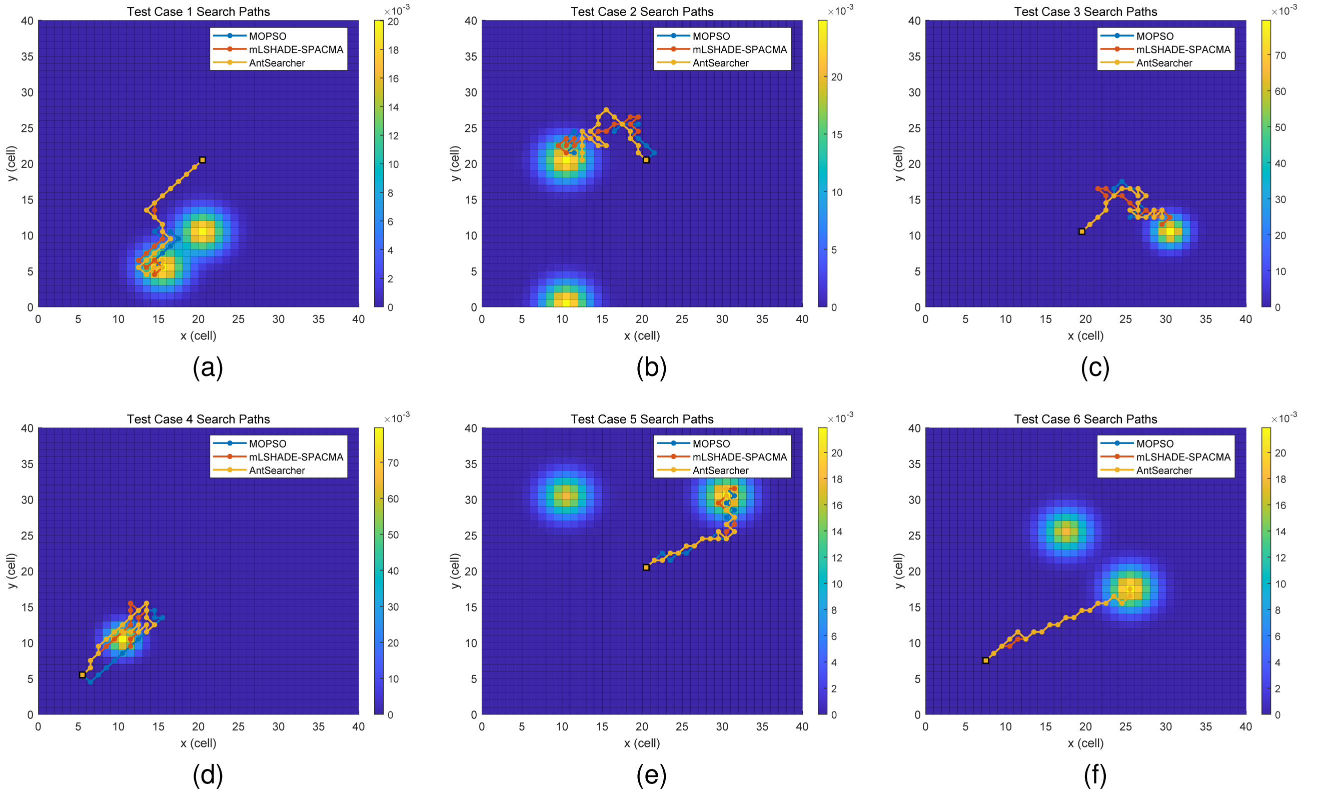
better-focused paths that effectively track the evolving high-probability regions, validating both the effectiveness of its dynamic bias mechanisms and its ability to escape local optima when necessary.

C. Effect of the telescope heuristic

To quantify the contribution of the two main proposed components, the Telescope heuristic and the dynamic heuristic decay, we perform a systematic ablation study.

Table II compares the full AntSearcher against AntSearcher-wo-Heur, a variant in which the Telescope heuristic tensor is removed (i.e., $\eta(i, j, k) \equiv 1$ for all cells and directions; only pheromone information guides the search). Surprisingly, the differences are not statistically significant in any scenario ($p > 0.05$). AntSearcher retains a slight edge in mean performance in five out of six cases, but the gaps are marginal. This indicates that, while the Telescope heuristic provides useful initial bias, the strong pheromone-based learning and restart mechanism are sufficient to discover high-quality solutions even in its absence. The heuristic appears most helpful in the early iterations for quickly directing ants toward promising regions, but its long-term impact is largely subsumed by pheromone reinforcement.

Table III compares the full AntSearcher against AntSearcher-FH (Fixed Heuristic), a variant that uses the original Telescope heuristic but disables the dynamic decay mechanism (i.e., the heuristic values remain constant throughout the run). The results clearly demonstrate the critical importance of gradually reducing heuristic influence: In five out of six scenarios (Cases 1, 2, 3, 5, and 6), AntSearcher



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 5. Best trajectories found by AntSearcher, MOPSO, and mLSHADE-SPACMA in six scenarios.

TABLE II
EFFECT OF THE TELESCOPE HEURISTIC

Case	AntSercher			AntSercher-wo-Heur				
	Best	Mean	Std	Best	Mean	Std	p-value	Sig.($p < 0.05$)
1	**0.20372**	**0.20240**	0.00050	0.20285	0.20188	0.00098	1.13E-01	Not Significant
2	**0.23721**	0.23263	0.00315	0.23703	**0.23275**	0.00355	8.53E-01	Not Significant
3	**0.68485**	**0.66787**	0.01220	0.68453	0.66450	0.01213	4.24E-01	Not Significant
4	**0.53737**	**0.50478**	0.01350	0.52745	0.49669	0.01631	1.34E-01	Not Significant
5	**0.21845**	**0.21489**	0.00367	**0.21845**	0.21339	0.00846	7.91E-01	Not Significant
6	0.16256	0.15219	0.01149	**0.16259**	**0.15621**	0.00915	6.80E-02	Not Significant

significantly outperforms AntSearcher-FH ($p \ll 0.05$), with mean detection probability gains ranging from 1.2% (Case 1) to as high as 38.7% relative improvement (Case 6). Only in Scenario 4 does AntSearcher-FH unexpectedly achieve a higher mean (0.52830 vs. 0.50478), possibly because the particular target dynamics and belief evolution in that instance temporarily align well with the static initial belief.

These results confirm that clinging to an outdated initial belief map is severely detrimental in most MTS problems. The dynamic decay mechanism successfully prevents the colony from becoming trapped in regions that were initially promising but later lose relevance due to target motion. By progressively shifting reliance from the (aging) heuristic to freshly updated pheromone trails, AntSearcher maintains adaptability and consistently achieves superior detection performance.

In summary, while the telescope heuristic offers moderate early guidance, the dynamic decay of heuristic influence is the key component responsible for AntSearcher's robustness across diverse and deceptive scenarios.

D. Effect of stagnation-triggered reinitialization

We now evaluate the contributions of the remaining two components: (i) the stagnation-detection and pheromone-reinitialization mechanism, and (ii) the adaptive updating of the initial pheromone level τ_0.

Table IV compares the full AntSearcher against AntSearcher-wo-Reinit, a variant that completely disables stagnation detection and pheromone reinitialization (i.e., the pheromone tensor is never reset after the initial setup). In three scenarios (Cases 1-3), the differences are not statistically significant, and AntSearcher-wo-Reinit occasionally posts marginally higher means. However, in the more deceptive Scenarios 4-6, removing the restart mechanism proves detrimental: AntSearcher significantly outperforms the non-restarting variant in Cases 4 ($p = 0.020$), 5 ($p = 0.017$), and

TABLE III
EFFECT OF THE DYNAMIC DECAY MECHANISM OF THE TELESCOPE HEURISTIC

Case	AntSercher			AntSercher-FH				
	Best	Mean	Std	Best	Mean	Std	p-value	Sig.($p < 0.05$)
1	**0.20372**	**0.20240**	0.00050	0.20235	0.20121	0.00129	7.50E-05	Better
2	**0.23721**	**0.23263**	0.00315	0.23639	0.22934	0.00362	9.64E-03	Better
3	**0.68485**	**0.66787**	0.01220	0.67838	0.65055	0.01548	1.40E-04	Better
4	0.53737	0.50478	0.01350	**0.56074**	**0.52830**	0.01095	5.96E-07	Worse
5	**0.21845**	**0.21489**	0.00367	0.21530	0.18711	0.01966	8.34E-07	Better
6	**0.16256**	**0.15219**	0.01149	0.15400	0.10969	0.02246	5.96E-08	Better

especially 6 ($p = 6.37 \times 10^{-5}$), where the mean detection probability drops from 0.15219 to 0.13101. These results confirm that stagnation-triggered reinitialization is crucial for escaping local optima in challenging scenarios involving multiple competing regions or rapidly evolving belief maps (particularly the diagonal corridor in Scenario 6).

Table V compares AntSearcher against AntSearcher-FT, which uses a fixed $\tau_0 = 0.2/m$ throughout the entire run instead of dynamically adjusting it based on the best-so-far detection probability. The full adaptive version exhibits statistically superior mean performance in the first three scenarios (Cases 1-3, $p < 0.05$), with gains of approximately 0.2-2.1% in detection probability. In the remaining three scenarios, the differences are not significant, suggesting that adaptive τ_0 is particularly beneficial when the solution landscape contains clear high-quality solutions early on (as in Scenarios 1-3). By scaling the reinitialized pheromone level to the quality of the current best tour, the adaptive strategy accelerates exploitation after each restart without excessively restricting diversity.

In summary, both mechanisms contribute meaningfully to AntSearcher's robustness. The stagnation-triggered reinitialization is essential for recovering from premature convergence in deceptive and highly dynamic scenarios, whereas adaptive τ_0 provides faster and more focused intensification in cases where strong solutions are discovered early. Their combination explains AntSearcher's consistent superiority over the baselines observed in the previous section.

V. CONCLUSION

This paper presents AntSearcher, a novel ant colony optimization algorithm specifically designed for the moving-target search (MTS) problem using a single UAV. By extending the classic Ant Colony System to a tensor-based representation on a discretized grid with eight possible actions per cell, AntSearcher effectively handles the dynamic and deceptive nature of the problem. Four key components distinguish the proposed approach: (1) the Telescope heuristic that exploits the initial belief map to provide informed early guidance, (2) a dynamic decay mechanism that gradually phases out the outdated heuristic as the target moves, (3) a stagnation-triggered pheromone reinitialization to escape local optima, and (4) an adaptive initial pheromone level τ_0 that scales with the quality of the best-so-far solution.

Extensive experiments on six benchmark scenarios demonstrate that AntSearcher consistently outperforms two state-of-the-art metaheuristics (MOPSO and mLSHADE-SPACMA) in detection probability. Ablation studies further confirm the importance of the dynamic heuristic decay and restart mechanisms, particularly in scenarios with multiple competing regions or rapidly evolving probability distributions.

We believe AntSearcher establishes a flexible foundation for ant-colony-based solutions to dynamic search and surveillance tasks, with potential for real-world deployment in search-and-rescue, wildlife monitoring, and security applications. Despite the advances, several promising directions remain for future research. First, it is possible to extend AntSearcher to multiple cooperating UAVs, including decentralized pheromone management and collision avoidance. Second, it is also critical to develop online replanning variants that can update the search trajectory in real time as new sensor measurements arrive.

REFERENCES

[1] C. O. Quero and J. Martinez-Carranza, "Unmanned aerial systems in search and rescue: A global perspective on current challenges and future applications," *International Journal of Disaster Risk Reduction*, p. 105199, 2025.

[2] Y. Alqudsi and M. Makaraci, "Uav swarms: research, challenges, and future directions," *Journal of Engineering and Applied Science*, vol. 72, no. 1, p. 12, 2025.

[3] M. Ishiwatari, "Leveraging drones for effective disaster management: a comprehensive analysis of the 2024 noto peninsula earthquake case in japan," *Progress in Disaster Science*, vol. 23, p. 100348, 2024.

[4] M. D. Phung and Q. P. Ha, "Motion-encoded particle swarm optimization for moving target search using uavs," *Applied Soft Computing*, vol. 97, p. 106705, 2020.

[5] K. Trummel and J. Weisinger, "The complexity of the optimal searcher path problem," *Operations Research*, vol. 34, no. 2, pp. 324–327, 1986.

[6] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of markov decision processes," *Mathematics of operations research*, vol. 27, no. 4, pp. 819–840, 2002.

[7] F. Bourgault, T. Furukawa, and H. F. Durrant-Whyte, "Optimal search for a lost target in a bayesian world," in *Field and Service Robotics: Recent Advances in Reserch and Applications*. Springer, 2006, pp. 209–222.

[8] M. Raap, S. Meyer-Nieberg, S. Pickl, and M. Zsifkovits, "Aerial vehicle search-path optimization: A novel method for emergency operations," *Journal of optimization theory and applications*, vol. 172, no. 3, pp. 965–983, 2017.

[9] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," *IEEE computational intelligence magazine*, vol. 1, no. 4, pp. 28–39, 2007.

[10] M. A. Awadallah, S. N. Makhadmeh, M. A. Al-Betar, L. M. Dalbah, A. Al-Redhaei, S. Kouka, and O. S. Enshassi, "Multi-objective ant colony optimization," *Archives of Computational Methods in Engineering*, vol. 32, no. 2, pp. 995–1037, 2025.

[11] J. Cui, L. Wu, X. Huang, D. Xu, C. Liu, and W. Xiao, "Multi-strategy adaptable ant colony optimization algorithm and its application in robot path planning," *Knowledge-Based Systems*, vol. 288, p. 111459, 2024.

[12] H. Heng, M. H. M. Ghazali, and W. Rahiman, "Exploring the application of ant colony optimization in path planning for unmanned surface vehicles," *Ocean Engineering*, vol. 311, p. 118738, 2024.

TABLE IV
EFFECT OF THE STAGNATION-DETECTION AND PHEROMONE-REINITIALIZATION MECHANISM

Case	AntSercher			AntSercher-wo-Reinit				
	Best	Mean	Std	Best	Mean	Std	p-value	Sig.($p < 0.05$)
1	0.20372	0.20240	0.00050	**0.20448**	**0.20241**	0.00114	5.53E-01	Not Significant
2	**0.23721**	**0.23263**	0.00315	0.23703	0.23209	0.00447	7.71E-01	Not Significant
3	**0.68485**	0.66787	0.01220	**0.68485**	**0.66976**	0.01026	2.75E-01	Not Significant
4	0.53737	0.50478	0.01350	**0.53738**	**0.51130**	0.01560	2.03E-02	Worse
5	**0.21845**	**0.21489**	0.00367	**0.21845**	0.19986	0.02005	1.73E-02	Better
6	**0.16256**	**0.15219**	0.01149	0.16032	0.13101	0.00747	6.37E-05	Better

TABLE V
EFFECT OF THE ADAPTIVE INITIAL PHEROMONE LEVEL τ_0

Case	AntSercher			AntSercher-FT				
	Best	Mean	Std	Best	Mean	Std	p-value	Sig.($p < 0.05$)
1	0.20372	**0.20240**	0.00050	**0.20448**	0.20205	0.00083	7.37E-03	Better
2	**0.23721**	**0.23263**	0.00315	0.23703	0.23039	0.00471	4.22E-02	Better
3	**0.68485**	**0.66787**	0.01220	0.67838	0.65355	0.01566	4.60E-03	Better
4	0.53737	**0.50478**	0.01350	**0.54162**	0.49848	0.02673	4.91E-01	Not Significant
5	**0.21845**	**0.21489**	0.00367	**0.21845**	0.21194	0.00874	9.03E-02	Not Significant
6	**0.16256**	**0.15219**	0.01149	0.16224	0.14865	0.01290	1.91E-01	Not Significant

[13] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia tools and applications*, vol. 80, no. 5, pp. 8091–8126, 2021.

[14] B. Alhijawi and A. Awajan, "Genetic algorithms: Theory, genetic operators, solutions, and applications," *Evolutionary Intelligence*, vol. 17, no. 3, pp. 1245–1256, 2024.

[15] M. A. Londe, L. S. Pessoa, C. E. Andrade, and M. G. Resende, "Biased random-key genetic algorithms: A review," *European Journal of Operational Research*, vol. 321, no. 1, pp. 1–22, 2025.

[16] M. N. Ab Wahab, A. Nazir, A. Khalil, W. J. Ho, M. F. Akbar, M. H. M. Noor, and A. S. A. Mohamed, "Improved genetic algorithm for mobile robot path planning in static environments," *Expert Systems with Applications*, vol. 249, p. 123762, 2024.

[17] M. Pant, H. Zaheer, L. Garcia-Hernandez, A. Abraham *et al.*, "Differential evolution: A review of more than two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 90, p. 103479, 2020.

[18] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE transactions on evolutionary computation*, vol. 15, no. 1, pp. 4–31, 2010.

[19] K. V. Price, "Differential evolution," in *Handbook of optimization: From classical to modern approach*. Springer, 2013, pp. 187–214.

[20] E. Reyes-Davila, E. H. Haro, A. Casas-Ordaz, D. Oliva, and O. Avalos, "Differential evolution: A survey on their operators and variants," *Archives of Computational Methods in Engineering*, vol. 32, no. 1, pp. 83–112, 2025.

[21] L. Abualigah, A. Sheikhan, A. M. Ikotun, R. A. Zitar, A. R. Alsoud, I. Al-Shourbaji, A. G. Hussien, and H. Jia, "Particle swarm optimization algorithm: review and applications," *Metaheuristic optimization algorithms*, pp. 1–14, 2024.

[22] D. Tian, Q. Xu, X. Yao, G. Zhang, Y. Li, and C. Xu, "Diversity-guided particle swarm optimization with multi-level learning strategy," *Swarm and Evolutionary Computation*, vol. 86, p. 101533, 2024.

[23] Y. Tang, K. Huang, Z. Tan, M. Fang, and H. Huang, "Multi-subswarm cooperative particle swarm optimization algorithm and its application," *Information Sciences*, vol. 677, p. 120887, 2024.

[24] Z. Zhang, J. Li, Z. Lei, Q. Zhu, J. Cheng, and S. Gao, "Reinforcement learning-based particle swarm optimization for wind farm layout problems," *Energy*, vol. 313, p. 134050, 2024.

[25] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4. IEEE, 1995, pp. 1942–1948.

[26] P. Lanillos, J. Yanez-Zuluaga, J. J. Ruz, and E. Besada-Portas, "A bayesian approach for constrained multi-agent minimum time search in uncertain dynamic domains," in *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, 2013, pp. 391–398.

[27] P. Lanillos, E. Besada-Portas, G. Pajares, and J. J. Ruz, "Minimum time search for lost targets using cross entropy optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 602–609.

[28] J. N. Eagle and J. R. Yee, "An optimal branch-and-bound procedure for the constrained path, moving target search problem," *Operations research*, vol. 38, no. 1, pp. 110–114, 1990.

[29] A. Sarmiento, R. Murrieta-Cid, and S. Hutchinson, "An efficient motion strategy to compute expected-time locally optimal continuous search paths in known environments," *Advanced Robotics*, vol. 23, no. 12-13, pp. 1533–1560, 2009.

[30] P. Lanillos, S. K. Gan, E. Besada-Portas, G. Pajares, and S. Sukkarieh, "Multi-uav target search using decentralized gradient-based negotiation with expected observation," *Information Sciences*, vol. 282, pp. 92–110, 2014.

[31] G. Mathews, H. Durrant-Whyte, and M. Prokopenko, "Asynchronous gradient-based optimisation for team decision making," in *2007 46th IEEE Conference on Decision and Control*. IEEE, 2007, pp. 3145–3150.

[32] C. Blum, "Ant colony optimization: A bibliometric review," *Physics of life reviews*, vol. 51, pp. 87–95, 2024.

[33] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, vol. 26, no. 1, pp. 29–41, 1996.

[34] M. Dorigo and L. M. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on evolutionary computation*, vol. 1, no. 1, pp. 53–66, 2002.

[35] T. Stützle and H. H. Hoos, "Max–min ant system," *Future generation computer systems*, vol. 16, no. 8, pp. 889–914, 2000.

[36] S. Fu, C. Ma, K. Li, C. Xie, Q. Fan, H. Huang, J. Xie, G. Zhang, and M. Yu, "Modified lshade-spacma with new mutation strategy and external archive mechanism for numerical optimization and point cloud registration," *Artificial Intelligence Review*, vol. 58, no. 3, p. 72, 2025.

[37] A. W. Mohamed, A. A. Hadi, A. M. Fattouh, and K. M. Jambi, "Lshade with semi-parameter adaptation hybrid with cma-es for solving cec 2017 benchmark problems," in *2017 IEEE Congress on evolutionary computation (CEC)*. IEEE, 2017, pp. 145–152.