Unmanned Systems, Vol. 14, No. 4 (2026) 717–730

DOI: 10.1142/S2301385026500147

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us

Unmanned Aerial Vehicle Trajectory Planning Technology Based on Improved ACO Fusion Algorithm

Hao Chen, Lankang Zhu*, Fa Wang, Yue Yao, Wei Han

Jiaxing Hengchuang Electric Power Design and Research Institute, Jiaxing 314033, P. R. China

With the advancement of contemporary science and technology, artificial intelligence has become a common practice across numerous industries, with unmanned aerial vehicles (UAVs) serving as its primary component. However, the current unmanned aerial vehicle trajectory planning is mostly limited to a single algorithm or a two-dimensional environment, with limited planning capability in a three-dimensional environment. Therefore, the study is to improve the ant colony algorithm and the artificial potential field method, respectively, and to combine the two improved algorithms to propose a fusion model, according to which the improved UAV trajectory planning (UAV-TP) system is designed. The outcomes indicated that the fusion model had the optimal mean and standard deviation on three different types of benchmark test functions, and possessed the best adaptability and stability. Applying the fusion algorithm to the experiments of the UAV-TP system, its optimal path in a simple 2D grid environment passed through 38 grids, produced 16 inflection points, 0 unsafe points, and searched for the optimal path at the 13th iteration. The average trajectory length in 20 experiments in a complex 3D environment was 27.59 km, which was 28.43% and 16.21% lower than the rest of the comparison algorithms, respectively, and the average number of corners was seven. The outcomes reveal that the fusion algorithm proposed by the institute performs well in both trajectory planning and safe navigation, and can realize efficient navigation in three-dimensional complex environments.

Keywords: Ant colony algorithm; artificial potential field method; unmanned aerial vehicle; trajectory planning; three-dimensional environment.

1. Introduction

Artificial intelligence (AI) technology has been extensively and in-depth researched in recent years with the goal of developing a strong scientific and technological nation and quickening the realization of high-level scientific and technological self-reliance [1]. Unmanned aerial vehicle (UAV) is an important part of AI technology, which is a kind of vehicle mainly controlled by radio remote control or by its own program [2, 3]. UAV has strong practical performance, due to its small size, safety and efficiency and convenience and sensitivity, it has a great degree of applicability in the field of disaster rescue, agricultural production and express logistics [4]. To achieve efficient mission execution, the trajectory planning system of UAV must be targeted and deeply designed [5]. In the past, UAV track planning systems were mostly focused on simple two-dimensional environments, and most of them used a single algorithm. They were unable to achieve fast and high-precision track planning in complex environments, and the target points were unreachable and easy to get stuck in local areas because of problems such as optimality and too many iterations. In recent years, research on UAV trajectory planning (UAV-TP) in three-dimensional environments has made significant progress. Researchers not only focus on two-dimensional environments, but also combine multiple algorithms for planning optimization in complex three-dimensional environments. For example, in addition, Fu *et al.* [6] proposed an optimization algorithm combining multi-agent collaboration for collaborative path planning of UAVs in high-threat environments. Experiments have shown that the algorithm can perform in complex three-dimensional spaces, producing faster convergence speed and higher

Received 6 August 2024; Accepted 7 February 2025; Published 22 March 2025. This paper was recommended for publication in its revised form by editorial board member, Yulong Ding.
Email Address: *p5s2y5@163.com
*Corresponding author.

stability. Therefore, the study improved the ant colony optimization (ACO) in four aspects, which are improving the initial pheromone distribution, limiting the path pheromone concentration (PC), and improving the heuristic function (HF) by setting the adaptive volatilization factor. The artificial potential field (APF) is introduced and its repulsive and gravitational functions are improved in the meantime to improve the algorithm's local planning effect. The improved APF is combined with ACO to propose the ant colony optimization-artificial potential field (ACO-APF) algorithm, which aims to aid in the advancement of AI technology. There are four sections to the research structure: Sec. 2 provides the review of relevant research findings. Section 3 provides the combination of enhanced ACO and improved APF algorithm is the proposed ACO-APF algorithm. Section 4 presents the analysis of UAV-TP and experimental results based on the enhanced algorithm. Section 5 includes the research summary and directions for further research.

The main contributions of this study are as follows: (1) A three-dimensional trajectory planning model ACO-APF is proposed that combines the improved ant colony algorithm and the APF method. By introducing an adaptive volatile factor and an improved HF, the algorithm's performance is improved. The global search capability and convergence speed, significantly reduce the risk of falling into local optimality during trajectory planning. (2) The effectiveness of the fusion model in complex environments is verified. The experimental results show that ACO-APF has significant advantages in terms of path length, number of turning points and safety, and is suitable for complex three-dimensional UAVs and trajectory planning requirements.

2. Related Works

ACO is widely used in many fields such as area construction, path planning and impact measurement, due to its good robustness and superior global search capability. Su *et al.* proposed a new collaborative motion planning method to improve the online collaboration and planning capabilities of autonomous vehicles. This method improves ACO and realizes multi-vehicle collaborative motion planning by generating multiple independent subpopulations and optimizing multi-objective functions. Simulation results show that this method is superior to the APF motion planning algorithm in terms of adaptability [7]. In order to meet the needs of improving the path planning and obstacle avoidance capabilities of mobile robots under the background of Industry 4.0, scholar Han J improved the node selection, pheromone update and transition probability calculation methods of ACO, and optimized the search direction and cost estimation method of the A* algorithm. Experiments show that the improved algorithm can effectively shorten the mobile path and training time, and has better convergence speed and minimum path length than traditional algorithms [8]. Gong *et al*. proposed a multi-trajectory planning model for the multi-trajectory planning problem of autonomous underwater vehicles in complex environments, and developed a trajectory optimizer based on ACO. The algorithm improves the search efficiency and solution diversity through niche strategies and diversified heuristic metrics. Experimental results show that while providing flexible trajectory selection, the algorithm can also outperform existing algorithms in single trajectory efficiency [9]. In order to solve the three-dimensional trajectory planning problem of UAVs in dynamic and complex environments, Sun *et al.* proposed a mixed integer programming model and designed an improved ACO by combining constraints, such as the risk of UAV crash, radio interference, and terrain obstacles. Numerical experimental results show that the improved algorithm can effectively reduce flight mileage and calculation time, and significantly improve the efficiency and safety of trajectory planning of UAVs in complex environments [10].

UAV is an important part of AI technology and has an important role in mapping, agriculture and logistics due to its adaptability, flexibility and security. Scholars such as Svaigen believed that jamming attack poses a great risk to UAV internet and affects the flight trajectory of UAV. Therefore, an airway-aware protection mechanism against jamming attacks on UAV internet was proposed to update the UAV path by analyzing the availability of airways. Experimental results demonstrated that the model is an effective protection mechanism with significant results in reducing interference and energy consumption [11]. Chinthi-Reddy and other scholars changed the privacy paradigm of UAV to prevent illegal collection of privacy information by UAV equipped with tracking and monitoring devices. To make it impossible for someone to find and track the UAV, the study randomized the route and obscured the UAV's current location. According to the experimental results, the model is a useful tool for tracking targets and partially protects trajectory and position privacy [12]. Zhang and other scholars proposed a path-oriented UAV-TP algorithm that can perform continuous and intensive image acquisition in the aerial path, and accurately incorporate the path quality and scene reconstruction quality into the optimization. The joint optimization of visual selection and path planning of UAV under this model was done simultaneously in one step. Results from experiments showed that this strategy can successfully lower the field acquisition cost of UAVs [13]. A new UAV trajectory prediction approach with an iterative future-past-socially compatible reflection module and a cascade sliding conditional variational automatic decoder module was proposed by Zhou and other scientists. It was capable of integrating the loss in the training of the pre-time step and reaching the loss minimization in the post-time step. Experimental results demonstrated the relatively superior performance of the method in trajectory prediction datasets [14].

In conclusion, prior trajectory planning techniques primarily relied on a single algorithmic model, despite the abundance of research findings on ACO and UAV-TP. As a result, the effectiveness of trajectory planning was significantly impacted by the constraints of this single model in complex situations. Therefore, a fusion model combining ACO and AFP is proposed, aiming to realize a highly accurate and applicable trajectory planning algorithm.

3. Design of UAV-TP System Based on ACO-APF Algorithm

The first part of the study improves the traditional ACO in terms of initial pheromone distribution, PC, HF, and adaptive factor. The second part optimizes the APF and proposes an ACO-APF fusion algorithm that combines the two.

3.1. *ACO algorithm improvements*

ACO is a classical probabilistic algorithm proposed by Dorigo based on the positive feedback process of the ant colony foraging process, after observing the traveling route of the ant colony [15]. The ACO algorithm provides a powerful tool for the solution of optimization problems and combinatorial optimization problems through the abstract modeling of the intelligent behavior of the colony, whose ant foraging behavior is schematically shown in Fig. 1.

In Fig. 1, ants will pick a path at random to get to the food source during the foraging phase. The ants that choose a shorter path at the same time will be the first to reach the food source and release pheromone, which will attract

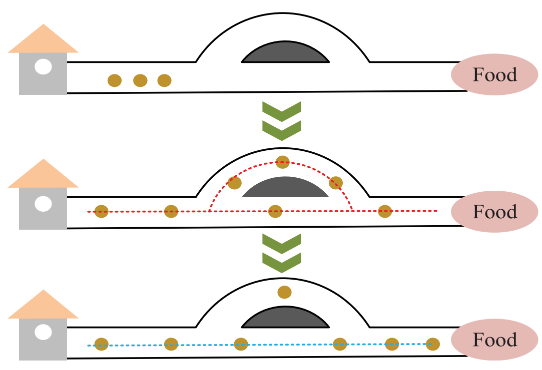


Fig. 1. Ant colony foraging behavior.

other ants to come to the shorter path, thus realizing the path searching optimization. It is vital to regulate the PC in a balanced manner, since ants will perpetually release and stockpile pheromones during the path optimization process [16]. In addition, nodes that have already been traversed are preferentially removed during path finding. The ACO algorithm mainly consists of probability and pheromone update. The transfer probability is calculated in the following equation [17]:

$$\begin{cases} P_{ij}^k(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^\alpha[\eta_{ij}(t)]^\beta}{\sum\limits_{j\in \text{allowed}_k}[\tau_{ij}(t)]^\alpha[\eta_{ij}(t)]^\beta}, & j \in \text{allowed}_k, \\ 0 & \text{otherwise}, \end{cases} \\ \eta_{ij}(t) = \dfrac{1}{c}. \end{cases} \tag{1}$$

In Eq. (1), $P_{ij}^k(t)$ is the probability that ant k passes through nodes i and j at moment t. $\tau_{ij}(t)$ is the PC between j nodes. α and β are the importance of PC and HF in probabilistic selection, respectively. allowed_k is the next set of nodes to be selected. $\eta_{ij}(t)$ is the HF and d_{ij} is the distance between i and j nodes. In the positive feedback process, the ants release pheromone to update the global pheromone after completing the cycle, the calculation is shown in the following equation [18]:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}(t), \\ \Delta\tau_{ij}(t) = \sum\limits_{k=1}^{n}\Delta\tau_{ij}^k(t), \end{cases} \tag{2}$$

where $\tau_{ij}(t+1)$ is the PC in the path. ρ is the volatilization coefficient of pheromone in the path. $\Delta\tau_{ij}(t)$ is the incremental sum of the pheromone at the moment t. $\Delta\tau_{ij}^k(t)$ denotes the PC released by the kth ant in the path of i and j nodes. Combined with the above calculations, the traditional ACO process in the initial search phase of the selection range is too wide, affecting the efficiency of the algorithm. So, the study will be based on the ACO added additional pheromone to avoid the blindness of the initial search, the calculation of which is shown in the following equation [19]:

$$\tau_0 = \tau_0'\left(1 + \frac{M}{d_{si} + d_{ie}}\right), \tag{3}$$

where τ_0' is the pheromone constant. M is the constant that controls the strength of the additional pheromone. d_{si} is the 3D space distance between node i and the starting point s. The distance d_{ie} is measured between the terminal point e and node i. More extra pheromone is attached if node i is closer to the connecting line ST from the beginning point s to the finishing point e, and less if node i is farther away. The PC of the location closer to the connecting line ST is set higher to enhance the targeting of the initial search. The study establishes a threshold for the PC, since the pheromone on the shorter paths will eventually fall into the local optimum as the number of repetitions grows. The following equation computes the pheromone limit [20]:

$$\tau(t+1)=\begin{cases}\tau_{\max}, & \tau(t)>\tau_{\max},\\ \tau(t), & \tau_{\min}\le\tau(t)\le\tau_{\max},\\ \tau_{\min}, & \tau(t)<\tau_{\min},\end{cases} \tag{4}$$

where $\tau_{\max}$ is the maximum value of PC. $\tau_{\min}$ is the minimum value of PC. This prevents local optimization due to too high PC and stagnation due to too low concentration. During the path search process, the HF is only the inverse of the distance between a node and the next node; using no objective point as a guideline will slow down the algorithm's convergence. An HF based on the angle of the straight line (SL) between nodes is set because of the UAV's performance limitations; Fig. 2 shows the angle of the SL schematically.

In Fig. 2, f is the previous node of the trajectory, i' is the current node, and j' is the next optional node. $\theta_{i'}$ is the angle in the SL where the line joins the three forms, and $s_{i'j'}$ denotes the direction factor. The $s_{i'j'}$ calculation is shown in the following equation:

$$s_{i'j'}=1-\cos\theta_{i'}. \tag{5}$$

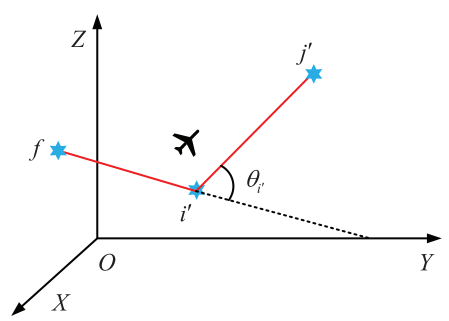


Fig. 2. Schematic diagram of the angle between straight lines.

In Eq. (5), the angle $\theta_{i'}$ of the formed SL is positively proportional to the direction factor $s_{i'j'}$. The study further considers the distance factor and the direction factor together for calculation, and assigns different weight coefficients to both of them, and the following equation computes the improved HF $\eta_{i'j'}$:

$$\begin{cases}\eta_{i'j'}=\dfrac{1}{a\times d_{j'e}+b\times s_{i'j'}},\\ d_{j'e}=\sqrt{(x_e-x_{j'})^2+(y_e-y_{j'})^2+(z_e-z_{j'})^2}\end{cases}. \tag{6}$$

In Eq. (6), $d_{j'e}$ denotes the distance factor, which is the distance from the to-be-selected node j' to the TarP e. $s_{i'j'}$ denotes the direction factor, and the weight coefficients of the direction and distance factors' relative degrees of influence are shown by the letters a and b, respectively, where $a+b=1$. $(x_{j'},y_{j'},z_{j'})$ are the coordinates of the to-be-selected node j', and (x_e,y_e,z_e) denotes the coordinates of the TarP e. The adaptive volatilization factor ρ' designed by the study is calculated in the following equation:

$$\begin{cases}\rho'(x=N_c)=R\dfrac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(N_c-\mu)^2}{2\sigma^2}},\\ \mu=\dfrac{1}{2}N_{\max},\\ \sigma=\dfrac{1}{4}N_{\max},\\ R=r\sqrt{2\pi}\sigma.\end{cases} \tag{7}$$

In Eq. (7), N_c denotes the current iteration and $N_{\max}$ denotes the maximum iteration. μ denotes the expected value, σ denotes the standard deviation. $\rho'\in[\rho'_{\min},\rho'_{\max}]$. R denotes a constant, and r denotes a fixed constant. Combined with the above calculations, the improved ACO process is shown in Fig. 3.

In Fig. 3, first, the 3D environment is modeled, second, the ant colony is initialized and the parameters and initial pheromone distribution are set, then the path search is performed and the next node is selected using the roulette

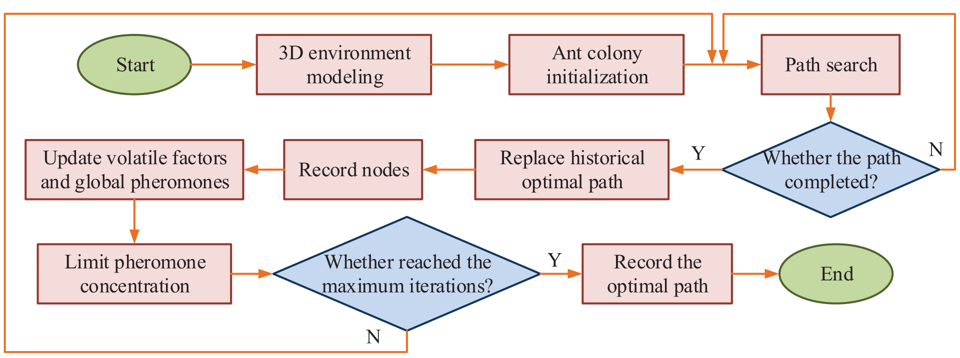


Fig. 3. Improved ACO algorithm flowchart.

method. Then, it is determined whether the ants complete the path or not, if they complete the path, then the route is kept and compared with the historical shortest path. If the route is a better one, then the historical shortest path is replaced and vice versa till the search is repeated. Subsequently, the nodes passed by the ants are recorded and the volatility factor and global pheromone are updated to limit the PC to the specified range. After determining whether the maximum iterations have been reached, the best path is noted and, if not, it is re-examined.

3.2. *Construction of ACO-APF fusion algorithm for UAV-TP*

To further improve the adaptability of the algorithm in obstacle avoidance and path optimization, the study introduced APF. By combining APF with ACO, the algorithm aims to improve its performance in dealing with obstacles, path smoothness, and local path optimization, in order to achieve more robust trajectory planning in complex environments. APF is an excellent local path planning method, which views the UAV, obstacles and TarPs as prime points, and models the flight area as a virtual potential field (PF). The TarP creates a gravitational pull on the UAV in the field of forces, directing it in its direction. In order to prevent the UAV from colliding, the barriers also exert repulsive force on it simultaneously. The gravitational and repulsive forces are schematically shown in Fig. 4.

The PF's gravitational and repulsive forces are schematically depicted in Fig. 4(a), respectively, while Fig. 4(b) depicts the force of the UAV in the PF. The gravitational force is stronger when the UAV is farther away from the TarP during the flight. The repulsive force increases with increasing distance between the UAV and the obstacle; as a result, the UAV will continue to move and adjust in response to both forces until it reaches the TarP. The following equation shows the calculation of the gravitational potential field (GPF), which is dependent on the distance between the UAV and the TarP:

$$\begin{cases} U_{\text{att}}(q) = \dfrac{1}{2} k_{\text{att}} d^2(q, q_g), \\ F_{\text{att}}(q) = -\nabla U_{\text{att}}(q) = -k_{\text{att}} d(q, q_g). \end{cases} \tag{8}$$

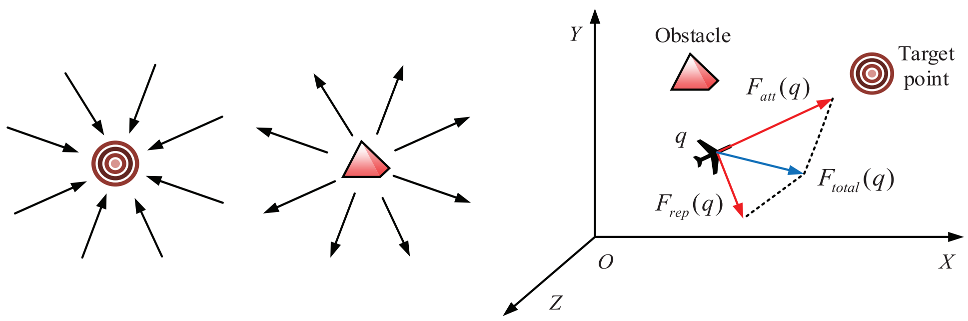
In Eq. (8), $U_{\text{att}}(q)$ is the GPF at the current position (CP) of the UAV and k_{att} is the GPF gain coefficient. q is the CP of the UAV and q_g is the target position. $d(q, q_g)$ is the distance from the UAV to the target and $F_{\text{att}}(q)$ is the gravitational force. The repulsive force is calculated in the following equation:

$$\begin{cases} U_{\text{rep}}(q) = \begin{cases} \dfrac{1}{2} k_{\text{rep}} \left(\dfrac{1}{d(q, q_0)} - \dfrac{1}{d_0} \right)^2, & d(q, q_o) \le d_0, \\ 0, & d(q, q_o) > d_0, \end{cases} \\ F_{\text{rep}}(q) = -\Delta U_{\text{rep}}(q) = \begin{cases} k_{\text{rep}} \left(\dfrac{1}{d(q, q_0)} - \dfrac{1}{d_0} \right) \dfrac{1}{d^2(q, q_0)}, & 0 \le d(q, q_0) \le d_0, \\ 0, & d(q, q_0) > d. \end{cases} \end{cases} \tag{9}$$

In Eq. (9), $U_{\text{rep}}(q)$ is the repulsive PF at the CP of the UAV and k_{rep} is the repulsive PF gain coefficient. q_0 is the obstacle position. The distance between the obstacle and the UAV is indicated by $d(q, q_o)$. d_0 is the maximum range at which the obstacle produces an effect, and $F_{\text{rep}}(q)$ is the repulsive force. Multiple obstacles can generate repulsive force at the same time, and the combined PF is calculated in the following equation:

$$\begin{cases} F_{\text{total}}(q) = \displaystyle\sum_{i=1}^{n} F_{\text{rep}}(q) + F_{\text{att}}(q), \\ U_{\text{total}}(q) = \displaystyle\sum_{i=1}^{n} U_{\text{rep}}(q) + U_{\text{att}}(q). \end{cases} \tag{10}$$

In Eq. (10), n is the total obstacles. Equation (11) illustrates the revised repulsive field computation. The work addresses the issue of target unreachability in the standard APF, by incorporating the UAV-to-target distance element



(a) Gravity (b) Repulsive force (c) UAV force diagram

Fig. 4. Artificial force field force diagram.

into the repulsive PF.

$$U_{\text{rep}}(q)=\begin{cases}\frac{1}{2}k_{\text{rep}}\left(\frac{1}{d(q,q_0)}-\frac{1}{d_0}\right)^2 d^{\lambda}(q,q_g), & d(q,q_g)\le d_0,\\ 0, & d(q,q_o)>d_0.\end{cases} \tag{11}$$

In Eq. (11), λ is a set constant with a positive value, and $d(q,q_g)$ is the distance from the UAV to the target. The improved repulsive force expression is shown in the following equation:

$$\begin{cases}F_{\text{rep}}(q)=-\nabla U_{\text{rep}}(q)=\begin{cases}F_{\text{rep1}}+F_{\text{rep2}}, & d(q,q_0)\le d_0,\\ 0, & d(q,q_0)>d_0,\end{cases}\\ F_{\text{rep1}}=k_{\text{rep}}\left(\frac{1}{d(q,q_0)}-\frac{1}{d_0}\right)\frac{1}{d^2(q,q_o)}d^{\lambda}(q,q_g),\\ F_{\text{rep1}}=-\frac{\lambda}{2}k_{\text{rep}}\left(\frac{1}{d(q,q_0)}-\frac{1}{d_0}\right)^2 d^{\lambda-1}(q,q_g).\end{cases} \tag{12}$$

In Eq. (12), F_{rep1} denotes the direction of the obstacle pointing toward the UAV and F_{rep2} denotes the direction of the UAV pointing toward the target. Therefore, the improved UAV forces are shown in Fig. 5.

The distance between the UAV and the target is introduced, and constant λ is set to two, as illustrated in Fig. 5. As the UAV is gradually approaching the target, F_{rep1} and F_{rep2} are gradually reduced to 0 in the process. Therefore, the combined force at the target position in the PF is the point of the global minimum, and the UAV can reach the end point. Meanwhile, the study introduces the escape force to increase the target's influence on the UAV to avoid the stagnation state, and the calculation is shown in the following equation:

$$\begin{cases}U_{\text{add}}(q)=\begin{cases}-\frac{1}{2}k_{\text{add}}\left(\frac{1}{d(q,q_0)}\right)^2, & d(q,q_0)\le d_a,\\ 0, & d(q,q_0)\le d_a,\end{cases}\\ F_{\text{add}}(q)=-\nabla U_{\text{add}}(q)=\frac{-k_{\text{add}}}{d^3(q,q_0)}.\end{cases} \tag{13}$$

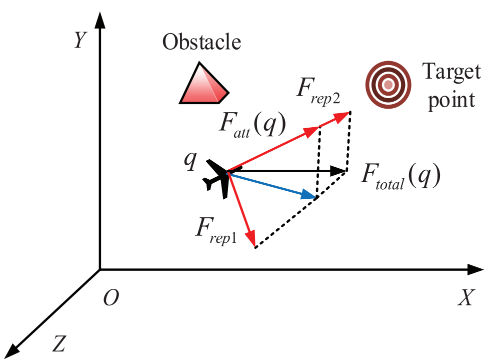


Fig. 5. Force diagram after improving repulsion.

In Eq. (13), $U_{\text{add}}(q)$ represents the introduced escape force PF, k_{add} represents the escape force coefficient, and d_a represents the escape force range. F_{add} represents the increased escape force by the UAV pointing in the direction of the target, at this time the UAV is subjected to three forces at the same time, and the PF force is shown in the following equation:

$$F_t(q)=F_{\text{att}}(q)+F_{\text{add}}(q)+\sum_{i=1}^{n}F_{\text{rep}}(q). \tag{14}$$

In Eq. (14), $F_t(q)$ denotes the improved PF force. Subsequently, the study introduces the improved APF force into the initial phase heuristic information of the ACO, whereby, the UAV is able to realize obstacle avoidance in the process of arriving at the target. After the improvement of APF is completed, the study combines the two and proposes the fusion algorithm ACO-APF. Based on this fusion algorithm, the improved HF is shown in the following equation:

$$\begin{cases}\eta_{ds}(t)=\eta_{i'j'}(t)\cdot\eta_F(t),\\ \eta_F(t)=c^{F_1\cos\varphi}.\end{cases} \tag{15}$$

In Eq. (15), $\eta_F(t)$ denotes the increased PF force heuristic factor and c denotes a constant with a positive value. φ denotes the angle between the SL direction of the UAV at the current node i' and the next to-be-selected node j' and the PF force F_t. More moderating factors are added to the iteration in order to decrease the impact of the PF later on, as the following equation illustrates:

$$\begin{cases}\delta=\frac{N_{\max}-N_c}{N_{\max}}\\ \eta_F(t)=\delta c^{F_t\cos\varphi},\\ \eta_{ds}(t)=\delta\frac{c^{F_t\cos\varphi}}{a\times d_{j'e}+b\times s_{i'j'}}.\end{cases} \tag{16}$$

In Eq. (16), δ denotes the PF regulation factor, for which the restrictions are given in the following equation:

$$\delta=\begin{cases}\frac{N_{\max}-N_c}{N_{\max}}, & \delta\ge\delta_{\min},\\ \delta_{\min} & \text{otherwise}.\end{cases} \tag{17}$$

In Eq. (17), $\delta_{\min}$ denotes the minimum value of the PF force regulation factor. Combining the above calculations, the heuristic factor and state transfer probability of ACO-APF are calculated in the following equation:

$$\begin{cases}\eta_{ds}=\begin{cases}\frac{N_{\max}-N_c}{N_{\max}}\cdot\frac{c^{F_t\cos\varphi}}{a\times d_{j'e}+b\times s_{i'j'}}, & \delta\ge\delta_{\min},\\ \delta_{\min}\cdot\frac{c^{F_t\cos\varphi}}{a\times d_{j'e}+b\times s_{i'j'}} & \text{otherwise}.\end{cases}\\ P_{i'j'}^{k}(t)=\begin{cases}\frac{[\tau_{i'j'}(t)]^{\alpha}[\eta_{ds}(t)]^{\beta}}{\sum_{j'\in\text{allowed}_k}[\tau_{i'j'}(t)]^{\alpha}[\eta_{ds}(t)]^{\beta}}, & j'\in\text{allowed}_k,\\ 0 & \text{otherwise}.\end{cases}\end{cases} \tag{18}$$

After the above calculation of HF, conditioning factor, and state probability, the flow of the proposed ACO-APF fusion algorithm of the institute is shown in Fig. 6.

In Fig. 6, first, the 3D environment modeling of the voyage is carried out, and second, the parameters of the ACO-APF algorithm are initialized and the starting and ending points are set. Subsequently, A ants are placed at the starting point and the regulation factor, PF force, and HF values of the PF are calculated. The ants are then used to search for the path and determine the next path node according to the roulette wheel method. The subsequent phase determines whether the ants have successfully reached the designated goal point. If this is the case, the path length is compared with the historical shortest path length. The previous optimal path (OP) is substituted if the present path is shorter. Rather, the path is re-examined and the force field's parameters are recalculated. Then, the information of the current shortest path nodes passed by the ants is recorded. The volatility factor is updated and a global pheromone update is performed to limit the PC to an appropriate range. Lastly, whether the maximum number of iterations has been achieved is ascertained. If achieved, the current best course of action and output of the outcome are noted., the ants are placed at the starting point and the above steps are repeated.

In order to ensure that the proposed ACO-AFP algorithm can operate safely and effectively in a real environment, the study also integrated a variety of dynamic constraints into the algorithm. These constraints are designed to improve the safety and efficiency of path planning while enhancing the algorithm's adaptability in complex environments. First, during the path generation process, the algorithm calculates the distance between the path point and the obstacle in real time. The minimum safety distance $d_{\min}$ is set to ensure that the distance $P(x, y, z)$ between each path point $O(x_o, y_o, z_o)$ and the nearest obstacle D satisfies the following equation:

$$D = \sqrt{(x - x_o)^2 + (y + y_o)^2 + (z - z_o)^2} > d_{\min}. \tag{19}$$

Then, to meet the mission requirements, the flight altitude range of the UAV is set to the minimum $h_{\min}$ and the maximum altitude $h_{\max}$. In path planning, the altitude z of each path point must satisfy the following equation:

$$h_{\min} < z < h_{\max}. \tag{20}$$

Finally, the algorithm dynamically controls the flight and sets the maximum speed $v_{\max}$. When generating the path, ensure that the speed v_i of each path segment satisfies the following equation:

$$v_i < v_{\max}. \tag{21}$$

The integration of the above dynamic constraints aims to make the ACO-AFP algorithm more robust and flexible in practical applications.

In the proposed ACO-AFP fusion algorithm, the time complexity of the algorithm is mainly determined by the calculation process of ACO and APF. First, the complexity of the ant colony algorithm depends on the population size, problem scale and number of iterations. In each iteration, each ant needs to traverse the entire search space to find the optimal path, so the time complexity of the ACO part is related to the population size M, the problem scale N and the number of iterations T. The time complexity of ACO is $O(M \times N \times T)$. Second, the APF method assists local path planning by calculating the gravitational and repulsive forces between the drone and the target point and obstacles. The PF force calculation complexity of each path node

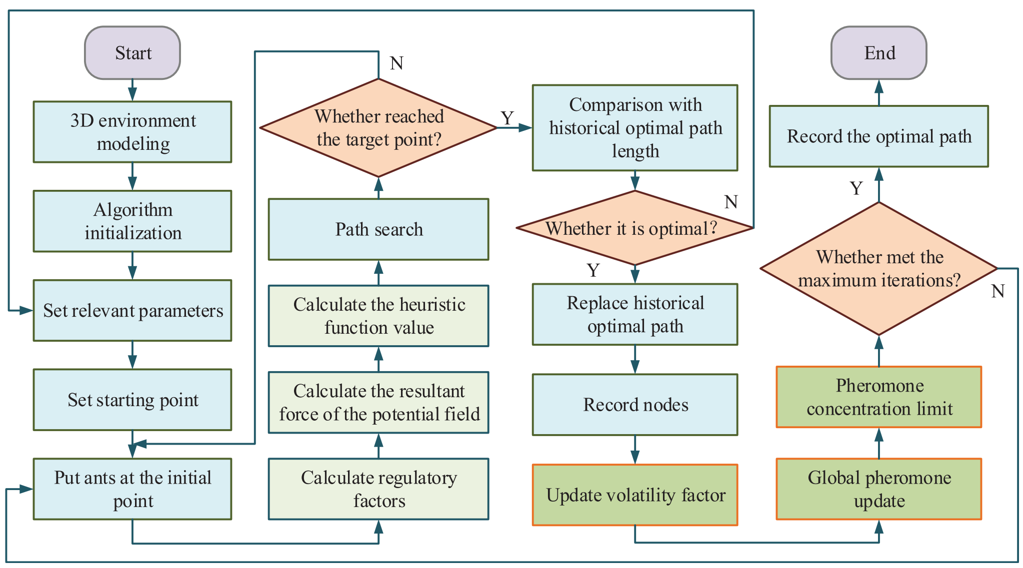


Fig. 6. ACO-APF algorithm flowchart.

is constant time $O(1)$, so the total complexity of the APF method is proportional to the number of nodes in the path N. The complexity of the APF method is $O(N)$.

After combining the two parts, the overall time complexity of ACO-AFP is mainly dominated by the ant colony algorithm, so the total complexity is $O(M \times N \times T)$. The complexity of the ACO-AFP fusion algorithm mainly depends on the problem scale N, the population size M and the number of iterations T. Although the algorithm is highly efficient in complex environments, it has a high degree of complexity and is therefore suitable for application scenarios that have certain requirements on computing resources.

4. Experiment and Result Analysis of UAV-TP System Based on ACO-APF Fusion Algorithm

In order to verify the feasibility of ACO-APF fusion algorithm in UAV-TP, the study will first test the performance of ACO-APF algorithm, and then conduct a comparative test on the effectiveness of the practical application of UAV-TP based on ACO-APF model. To guarantee the experiment's accuracy, all comparison algorithms use the same dataset and set of parameters.

4.1. *ACO-APF algorithm performance comparison test*

The study is conducted using a desktop computer with Windows 10 operating system running on 16 GB of RAM with Intel(R) Core (TM) i5-9400F processor. In order to evaluate the impact of different parameters in the ACO-AFP algorithm on its performance, a detailed parameter sensitivity analysis was conducted. Five key parameters were selected, namely population size, number of iterations, pheromone volatility coefficient, heuristic information weight, and pheromone weight. By adjusting each parameter one by one and keeping other parameters unchanged, the optimal path length and calculation time under different parameter settings were recorded to observe their impact on the algorithm performance. The experimental results are shown in Table 1.

Table 1. Sensitivity analysis results for ACO-AFP algorithm.

Parameter	Value	Convergence speed (iterations)	Stability (standard deviation)	Average fitness value
Population size (M)	20	95	0.031	−1535.45
	40	82	0.043	−1611.34
	60	90	0.029	−1698.25
	80	75	0.025	−1782.11
	100	85	0.017	−1856.43
Iterations (T)	100	80	0.027	−1621.67
	200	72	0.033	−1655.21
	300	88	0.015	−1723.14
	400	73	0.041	−1825.43
	500	89	0.019	−1922.54
Evaporation coefficient (ρ)	0.1	86	0.020	−1515.90
	0.3	77	0.018	−1625.37
	0.5	88	0.014	−1725.64
	0.7	79	0.032	−1801.89
	0.9	68	0.041	−1921.23
Heuristic weight (α)	0.5	84	0.016	−1525.45
	1	78	0.027	−1621.89
	1.5	91	0.029	−1732.43
	2	87	0.032	−1805.32
	2.5	81	0.02	−1913.64
Pheromone weight (β)	1	90	0.017	−1535.54
	1.5	79	0.021	−1625.67
	2	75	0.036	−1727.55
	2.5	68	0.03	−1825.42
	3	82	0.022	−1915.31

As can be seen from the table, as the population size increases, the convergence speed of the algorithm gradually increases, but the gain effect weakens after $M > 80$. A larger population makes the algorithm more stable and the standard deviation decreases. The increase in the number of iterations also accelerates convergence, but the effect is no longer obvious after more than 400 times. The volatility coefficient and heuristic information weight have a great influence on the algorithm. Moderate settings balance the convergence speed and result quality, while extreme settings lead to performance fluctuations. According to the parameter sensitivity analysis, the parameters finally selected are population size 80, number of iterations 400, volatility coefficient 0.5, heuristic weight 1.5, and pheromone weight 2.0.

The algorithms are first compared in single modal benchmark function (SMBF), multimodal benchmark function (MMBF) and fixed-dimensional multimodal benchmark function (FDMMBF). The algorithm's capacity to converge and be exploited is determined using SMBF. MMBF is employed to identify instances of exploration and algorithmic jumping out of the local optimum. FDMMBF is a combined function of SMBF and MMBF, which is usually used to detect the performance of the algorithm between exploration-exploitation. For ease of writing, the three functions are denoted as F_1 [21], F_2 [22], and F_3 [23] functions, and the specific information of the three different types of functions selected is depicted in Table 2.

Table 2 shows the selected functions and the corresponding information including expressions, dimensions,

Table 2. Detailed information of test functions.

Function	Formula	Dimension	Variable value range	Theoretical minimum value
F_1	$F_1(x) = \sum_{i=1}^{D} x_i^2$	30	$[-100, 100]$	0
F_2	$F_2(x) = \sum_{i=1}^{D} -x_i \sin(\sqrt{\lvert x_i \rvert})$	30	$[-500, 500]$	$-418.9829 \times D$
F_3	$F_3(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}\right]^2$	4	$[-5, 5]$	0.0003

intervals and theoretical minima. The study introduces ACO [24], maximum-minimum ant colony algorithm (MMAS) [25] for performance comparison test with the proposed ACO-AFP fusion algorithm. Each function is run 20 times independently. Table 3 displays the mean and standard deviation of each method following testing.

As shown in Table 3, the ACO-AFP algorithm proposed by the Institute achieves the optimal mean and standard deviation on all the functions. In the unimodal test function $F1$, ACO-AFP reached the theoretical optimal value, with an average value and standard deviation of 0.00, which was significantly better than ACO and MMAS, showing extremely high stability and accuracy. In the multimodal test function $F2$, the average value of ACO-AFP is -12569.48, far exceeding ACO's -7856.27 and MMAS's -5279.58, and the standard deviation is lower, indicating that the algorithm can jump out of the local optimum and maintain the consistency of results. The ability is stronger. In the mixed dimension test function F3, the average value of ACO-AFP is 3.21e-4, which is better than other algorithms, and its standard deviation is only 1.07e-19, which further reflects the global search capability and stability of the algorithm. The mean and standard deviation of the ACO-AFP algorithm on both MMBF and FDMMBF are minimized among the compared functions. This shows that the ACO-AFP algorithm has a clear advantage on the SMBF where only one global optimum exists, has the ability to get rid of the local optimum on multimode functions where a large number of local optimal solutions exist, and shows good exploration-exploitation switching ability on the fixed-dimension MMBF. Subsequently, the box plots of the fitness of each algorithm on the three functions are shown in Fig. 7.

Table 3. Mean and standard deviation results.

Function	Index	ACO	MMAS	ACO-AFP
F_1	Average	4.89e-105	2.23e-42	0.00
	Standard deviation	2.70e-104	1.37e-41	0.00
F_2	Average	-7856.27	-5279.58	-12569.48
	Standard deviation	3.66e3	4.83e2	9.83e-3
F_3	Average	5.27e-4	4.99e-2	3.21e-4
	Standard deviation	1.96e-4	3.85e-2	1.07e-19

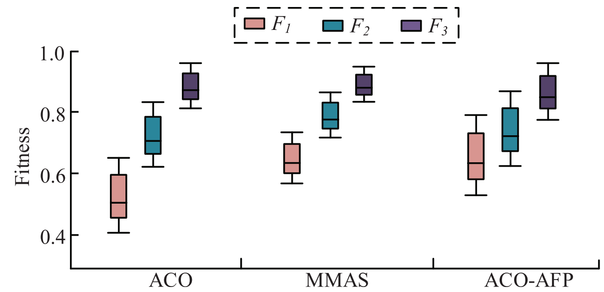


Fig. 7. Fitness diagram of three algorithms.

Since the maximum value of the function is used as the objective, the size of each individual's algorithmic fitness in Fig. 7 influences their individual merit, and the value of the objective function can be used directly to determine an individual's fitness. The figure illustrates the relatively narrow box plots on various functions that the suggested ACO-AFP algorithm of the study possesses, as well as the significant consistency of the ACO-AFP algorithm with respect to the maximum, minimum, and median values of the three functions. The fitness of MMAS is lower than the ACO-AFP algorithm, but slightly better than the traditional ACO algorithm. The stability of ACO's fitness on different types of functions is poor, and the ACO-AFP algorithm presents excellent stability on different types of functions.

Finally, the study will introduce three variants of ACO, namely MMAS, Elite Ant Colony System (EACO) [26], Standard Ant Colony Algorithm (SACO) [27], and three similar algorithms combining ACO with APF, namely Artificial Potential Field-Ant Colony Optimization (APF-ACO) [28], Optimized Artificial Potential Field-Ant Colony Optimization (Optimized APF-ACO) [29], and Improved Artificial Potential Field-Ant Colony Optimization (Improved APF-ACO) [30] as comparison algorithms to test the performance of the ACO-AFP fusion model proposed by the study. The experiment designed an algorithm performance test in a static environment. The experiments were conducted in test scenarios of different sizes, including 30×30, 100×100, and 500×500 static obstacle mazes, mainly testing the performance of path planning. The experimental results are shown in Table 4.

As can be seen from the table, the proposed model performs well in many indicators. In the 30×30 scenario,

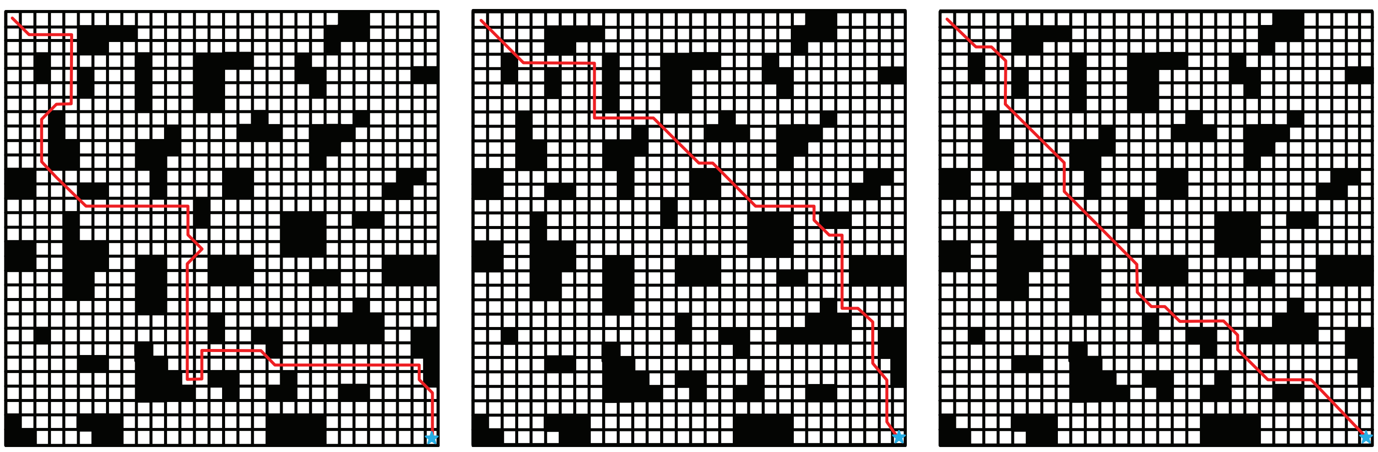
Table 4. Comparison of algorithm in different scaled path planning environments.

Algorithm	Scale	Iterations	Fitness	Time complexity (s)	Memory cost (MB)
MMAS [25]	30×30	148.2	0.891	6.03	44.5
EACO [26]		137.6	0.913	5.63	46.9
SACO [27]		182.4	0.875	7.05	49.3
APF-ACO [28]		129.5	0.925	5.25	42.8
Optimized APF-ACO [29]		120.7	0.946	5.03	41.6
Improved APF-ACO [30]		114.3	0.958	4.47	40.3
Proposed model		115	0.955	4.85	39.9
MMAS [25]	100×100	297.8	0.917	12.18	51.0
EACO [26]		281.3	0.931	11.58	52.6
SACO [27]		344.9	0.887	14.75	54.9
APF-ACO [28]		267.5	0.952	10.56	48.3
Optimized APF-ACO [29]		252.1	0.961	9.98	46.8
Improved APF-ACO [30]		232.9	0.973	9.34	46.0
Proposed model		235.4	0.965	9.40	45.7
MMAS [25]	500×500	497.6	0.879	25.48	58.7
EACO [26]		481.9	0.908	24.78	60.9
SACO [27]		549.8	0.862	27.96	64.8
APF-ACO [28]		463.4	0.915	22.34	54.7
Optimized APF-ACO [29]		429.1	0.937	21.03	52.8
Improved APF-ACO [30]		405.6	0.948	20.09	50.4
Proposed Model		410.6	0.944	20.38	49.8

the model iteration number is 115 times, close to the 114.3 times of the improved APF-ACO, and the fitness reaches 0.955, and the memory consumption is 39.9 MB, which is lower than most of the comparison algorithms. In the 100×100 scenario, the number of iterations is 235.4 times, the fitness is 0.965, slightly better than the 0.961 of the optimized APF-ACO, and the time complexity is 9.40 s. In the 500×500 scenario, the fitness is 0.944, the number of iterations is 410.6 times, and the memory consumption is kept at 49.8 MB, showing good resource utilization.

4.2. *Experimental analysis of UAV-TP based on ACO-APF algorithm*

On the basis of verifying the performance of the ACO-AFP algorithm, the ACO-AFP algorithm is further applied to the UAV-TP model test to verify its application capability. The study designed a spatial simulation environment for both simple and complex environments, which was first tested on a simple environment with a 30×30 grid, and the results are shown in Fig. 8.



(a) ACO (b) MMAS (c) ACO-AFP

Fig. 8. Paths of various models in a simple environment.

Figures 8(a)–8(c) display the OPs generated by the three algorithms ACO, MMAS, and ACO-AFP, respectively, after algorithmic iterations when applied to the UAV-TP system in a simple environment. In the simple environment, all the algorithms reach the goal point, however, the routes of the algorithms differ. To make the comparison clearer, the optimal path length, number of grids passed, number of turning points generated, and number of unsafe points passed of different algorithms are presented in tabular form, see Table 5.

As shown in Table 5, the optimal path length of the ACO-AFP algorithm is 27.59 m, which is significantly shorter than ACO's 38.57 m and MMAS's 32.92 m. The OP of the ACO algorithm passes through a total of 60 grids and generates 19 inflection points (IP). The OP of the MMAS algorithm passes through a total of 46 grids and generates 17 IPs. The OP of the ACO-AFP algorithm passes through only 38 grids and generates 16 IPs. In addition, the optimal trajectories of the ACO and MMAS models have five and three unsafe points, respectively, i.e. the paths have intersection points with black obstacles. The optimal trajectory of the ACO-AFP model, on the other hand, has no unsafe points, which not only minimizes the number of grids the optimal trajectory passes through but also has the shortest path to reach the TarP and can effectively implement safe obstacle avoidance. The iterative convergence of the algorithms of the three models in the above experiments is shown in Fig. 9.

Table 5. Performance comparison of different algorithms in path planning.

Metric	ACO	MMAS	ACO-AFP
Optimal path length (m)	38.57	32.92	27.59
Number of grids passed	60	46	38
Number of inflection points	19	17	16
Number of unsafe points passed	5	3	0

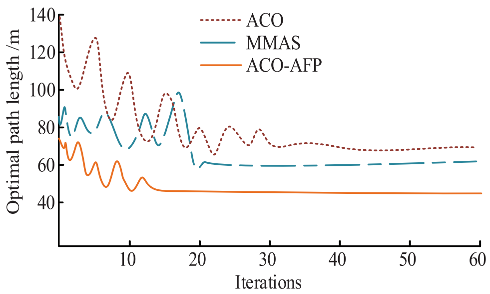
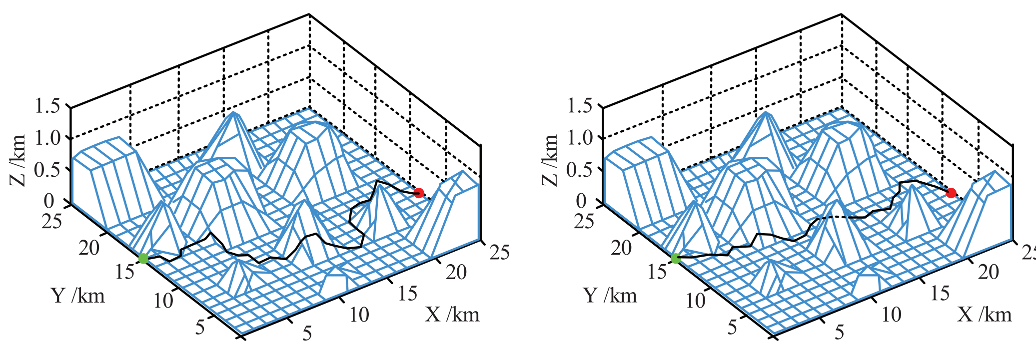


Fig. 9. Iterative convergence graph in a simple environment.

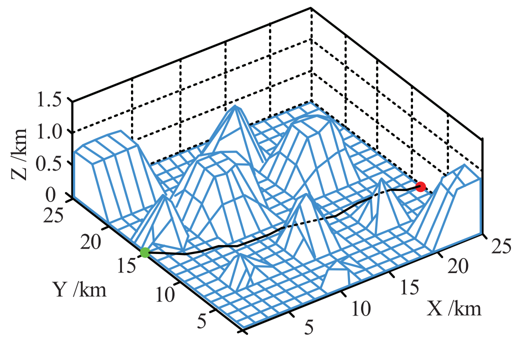
Figure 9 displays the iterations of the algorithm for each model in the simple environment. The ACO model searched for the historical OP at 39 iterations. The MMAS model searches for the historical OP at 22 iterations. The ACO-AFP model searches the historical OP only at 13 iterations. In addition, the average values of the historical paths of the three models ACO, MMAS, and ACO-AFP are 83.57, 57.9, and 46.78. The ACO-AFP model has the least number of iterations and has a small oscillation of the curve in the pre-iteration period. After the experiments in the simple environment, the study will establish a more complex three-dimensional space environment with a change in the location of the start and end points. The coordinates of the starting point (0, 15, 1), the coordinates of the TarP (25, 7, 1), and the remaining parameters are kept unchanged. The three-dimensional space has dimensions of 25 by 25 by 15. Figure 10 displays the findings of a comparison of 20 simulation experiments involving the three models in the complex environment.

Figures 10(a)–10(c) show the optimal trajectory diagrams of the three models ACO, MMAS, and ACO-AFP, respectively, after 20 trials in the complex environment. According to the statistical results, the ACO-AFP model has the shortest trajectory path line with an average of 27.59 km and an average number of corners of 7 times. The mean value of the trajectory path line of the ACO model is



(a) ACO (b) MMAS

Fig. 10. Each model in three-dimensional space.



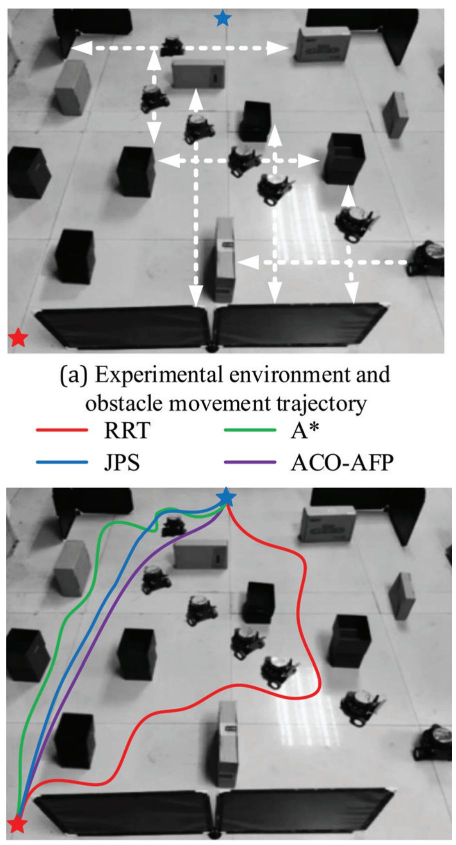
(c) ACO-AFP

Fig. 10. (*Continued*)

38.57 km, and the average number of corners is 15 times. The mean value of the trajectory path line of the MMAS model is 32.92 km, and the average number of corners is 11 times. In addition, the ACO-AFP algorithm did not produce any unsafe points in 20 experiments, while the ACO and MMAS algorithms passed five and three unsafe points, respectively, indicating that ACO-AFP has significant advantages in obstacle avoidance safety. The optimal trajectory planned by the ACO-AFP model is the shortest, and the trajectory is smoother, which is in line with the flight reality of UAV, and the trajectory planning effect is better in the same simulation environment.

Finally, in order to verify the feasibility of the deployment of the ACO-AFP algorithm in a real environment and show its difference from the classic path planning algorithms, namely Rapidly-exploring Random Tree (RRT) [31], A-star Algorithm (A*) [32] and Jump Point Search (JPS) [33], the study conducted simulation experiments on the ROS (Robot Operating System) platform. The experiment used ROS Melodic and Gazebo 9 simulation tools. As the official simulation platform of ROS, Gazebo can provide real physical engine and sensor simulation capabilities to ensure the authenticity and reliability of the experimental results. On the hardware platform, the simulation used the TurtleBot3 robot, which is equipped with a Light Detection and Ranging (LIDAR) sensor and an inertial measurement unit (IMU), which can collect environmental information in real time and perform path planning. In the experiment, each algorithm was encapsulated into the ROS navigation package and run on the robot's control module. Multiple physical obstacles were set up in the experimental environment to simulate a complex environment. The experimental site is a 5×5 m rectangular area, where the obstacle moves along a fixed path. The robot needs to plan from the starting point to the end point, avoiding these obstacles, and demonstrating the algorithm's response and obstacle avoidance capabilities in a real physical environment. The moving speed of TurtleBot3 is set to 0.5 m/s, and images are collected once per second during the experiment to generate a path map. The experimental site is shown in Fig. 11.

Figures 11(a) and 11(b) are the experimental site settings and the path planning results of each algorithm, respectively. As can be seen from Fig. 11(b), RRT generates an irregular path with a length of 7.8 m due to its fast-expanding random tree characteristics. The planning time is



(a) Experimental environment and obstacle movement trajectory

(b) Path planning results

Fig. 11. Experimental site map and experimental results under ROS.

1.34 s, including 10 turning points, and the path is long and not smooth enough. A* uses the heuristic search to plan a path along the edge of the obstacle, and the path length is shortened to 6.2 m. However, due to the conservative obstacle avoidance strategy, the number of turning points reaches nine and the planning time is 1.67 s. JPS skips redundant nodes through hopping search and generates a more straight path with a length of 5.7 m, only seven turning points, and a planning time of 1.52 s. In contrast, ACO-AFP uses the pheromone guidance mechanism to generate an optimal path with a length of only 5.3 m, the number of turning points is reduced to five, and the planning time is 1.42 s, showing excellent path smoothness and obstacle avoidance capabilities.

5. Conclusion

Conventional intelligent algorithms in UAV-TP systems typically have issues with easily falling into local minima and unachievable goals. The study combines the improved ACO algorithm with APF and proposes an ACO-AFP fusion model to realize safe and efficient UAV-TP. The outcomes indicated that the proposed ACO-AFP algorithm of the study took the optimal mean and standard deviation on SMBF, MMBF and FDMMBF and reached the theoretical optimal value of 0.00×10^0 on SMBF. In the fitness test, the ACO-AFP algorithm presented relatively narrower box plots than the rest of the algorithms on different functions, and had the best consistency in terms of maximum, minimum, and median in different functions. Second, applying the algorithm to the simple environment of the UAV-TP system, the OP of ACO-AFP passed through 38 grids, producing 16 IPs and 0 unsafe points. ACO, MMAS, and ACO-AFP searched for the historical OP at the 39th, 22nd, and 13th iterations, respectively. In the complex 3D environment, the average trajectory length of the ACO-AFP model was 27.59 km, and the average number of corners was 7. This suggests that, in comparison to other models, the study's suggested model performs better when it comes to UAV-TP. Since the experiments are only simulated in the simulated 3D maps and lack of testing in the outdoor field, it is expected that the improved algorithm will be physically experimented and optimized in the future when available.

ORCID

Hao Chen https://orcid.org/0000-0001-5928-3795
Lankang Zhu https://orcid.org/0009-0001-3925-9410
Fa Wang https://orcid.org/0009-0001-2985-7097
Yue Yao https://orcid.org/0009-0004-5421-0748
Wei Han https://orcid.org/0009-0004-1987-3722

References

[1] M. Hasanvand, M. Nooshyar, E. Moharamkhani and A. Selyari, Machine learning methodology for identifying vehicles using image processing, *Artif. Intell. Appl.* **4**(1) (2023) 170–178.
[2] Q. Qiu and D. Lau, Real-time detection of cracks in tiled sidewalks using YOLO-based method applied to unmanned aerial vehicle (UAV) images, *Autom. Constr.* **147**(3) (2023) 2–15.
[3] R. Zhang, G. Hao, K. Zhang and Z. Li, Unmanned aerial vehicle navigation in underground structure inspection: A review, *Geol. J.* **58**(6) (2023) 2454–2472.
[4] A. Pons and F. Cirak, Quaternion variational integration for inertial maneuvering in a biomimetic unmanned aerial vehicle, *ASME Lett. Dyn. Syst. Control* **3**(1) (2023) 2–9.
[5] S. B. Lee, D. Han and M. Song, Calculation and comparison of earthwork volume using unmanned aerial vehicle photogrammetry and traditional surveying method, *Sens. Mater.* **34**(12) (2022) 4737–4753.
[6] Y. Fu, S. Yang, B. Liu, E. Xia and D. Huang, Multi-UAV cooperative trajectory planning based on the modified cheetah optimization algorithm, *Entropy* **25**(9) (2023) 1277.
[7] S. Su, X. Ju, C. Xu and Y. Dai, Collaborative motion planning based on the improved ant colony algorithm for multiple autonomous vehicles, *IEEE Trans. Intell. Transp. Syst.* **25**(3) (2023) 2792–2802.
[8] J. Han, Application of improved ant colony algorithm integrating adaptive parameter configuration in robot mobile path design, *Int. J. Adv. Comput. Sci. Appl.* **14**(8) (2023) 397–407.
[9] Y. J. Gong, T. Huang, Y. N. Ma, S. W. Jeon and J. Zhang, MTrajPlanner: A multiple-trajectory planning algorithm for autonomous underwater vehicles, *IEEE Trans. Intell. Transp. Syst.* **24**(4) (2023) 3714–3727.
[10] B. Sun, J. Song and M. Wei, 3D trajectory planning model of unmanned aerial vehicles (UAVs) in a dynamic complex environment based on an improved ant colony optimization algorithm, *J. Nonlinear Convex Anal.* **25**(4) (2024) 737–746.
[11] A. Svaigen, A. Boukerche, L. Ruiz and A. Loureiro, Trajectory matters: Impact of jamming attacks over the drone path planning on the internet of drones, *Ad Hoc Netw.* **146**(7) (2023) 2–16.
[12] S. Chinthi-Reddy, S. Lim, G. Choi, J. Chae and G. Pu, DarkSky: Privacy-preserving target tracking strategies using a flying drone, *Veh. Commun.* **35**(6) (2022) 2–13.
[13] H. Zhang, Y. Yao, K. Xie, C. Fu, H. Zhang and H. Huang, Continuous aerial path planning for 3D urban scene reconstruction, *ACM Trans. Graph.* **40**(6) (2021) 2–16.
[14] H. Zhou, X. Yang, D. Ren, H. Huang and M. Fan, CSIR: Cascaded sliding CVAEs with iterative socially-aware rethinking for trajectory prediction, *IEEE Trans. Intell. Transp. Syst.* **24**(12) (2023) 14957–14969.
[15] 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)* **26**(1) (1996) 29–41.
[16] M. Scianna, The AddACO: A bio-inspired modified version of the ant colony optimization algorithm to solve travel salesman problems, *Math. Comput. Simul.* **218**(4) (2024) 357–382.
[17] G. Yuelin, W. Hongguang and W. Wanting, A hybrid ant colony optimization with fireworks algorithm to solve capacitated vehicle routing problem, *Appl. Intell.* **53**(6) (2023) 7326–7342.
[18] C. Nemade and U. Pujeri, Emergency automobile data transmission with ant colony optimization (ACO), *J. Adv. Inf. Technol.* **15**(5) (2023) 1003–1011.
[19] L. Xie, X. Li and X. Wei, A two-layer heterogeneous ant colony system with applications in part-picking planning, *J. Comput. Inf. Sci. Eng.* **23**(2) (2023) 2–13.

[20] G. A. Sgarro and L. Grilli, Ant colony optimization for Chinese postman problem, *Neural Comput. Appl.* **36**(6) (2024) 2901–2920.

[21] M. Laguna and R. Marti, Experimental testing of advanced scatter search designs for global optimization of multimodal functions, *J. Global Optim.* **33** (2005) 235–255.

[22] H. P. Schwefel*Numerical Optimization of Computer Models* (John Wiley & Sons, Inc., 1981).

[23] H. Rosenbrock, An automatic method for finding the greatest or least value of a function, *Comput. J.* **3**(3) (1960) 175–184.

[24] M. Dorigo, V. Maniezzo and A. Colorni, The ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. B, Cybern.* **26**(1) (1996) 29–41.

[25] T. Stützle and H. H. Hoos, MAX-MIN ant system, *Future Gener. Comput. Syst.* **16**(8) (2000) 889–914.

[26] M. Dorigo and G. Di Caro, The ant colony optimization metaheuristic, in *New Ideas in Optimization*, eds. D. Corne, M. Dorigo and F. Glover (McGraw Hill, London, UK, 1999), pp. 11–32.

[27] M. Dorigo and L. M. Gambardella, Ant colonies for the traveling salesman problem, *Biosystems* **43**(2) (1997) 73–81.

[28] J. Fu, T. Lv and B. Li, Underwater submarine path planning based on artificial potential field ant colony algorithm and velocity obstacle method, *Sensors* **22**(10) (2022) 3652.

[29] M. Li, B. Li, Z. Qi, J. Li and J. Wu, Optimized APF-ACO algorithm for ship collision avoidance and path planning, *J. Mar. Sci. Eng.* **11**(6) (2023) 1177.

[30] G. Chen and J. Liu, Mobile robot path planning using ant colony algorithm and improved potential field method, *Comput. Intell. Neurosci.* **2019**(1) (2019) 1932812.

[31] S. M. Lavalle, Rapidly-exploring random trees: A new tool for path planning, Technical Report **98**(11) (1998) 1–6.

[32] P. E. Hart, N. J. Nilsson and B. Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Trans. Syst. Sci. Cybern.* **4**(2) (1968) 100–107.

[33] D. Harabor and A. Grastien, Online graph pruning for pathfinding on grid maps, in *AAAI Conference on Artificial Intelligence* 25(1) (2011) 1114–1119.

Hao Chen obtained a Bachelor's degree in Communication Engineering from Donghua University in 2005. He obtained a Master's degree in Electrical Engineering from Zhejiang University in 2013. Currently, he is working as a senior engineer at Jiaxing Hengchuang Electric Power Design and Research Institute. His areas of interest are computer communication technology, intelligent networks and network security, and smart grids.

Yue Yao obtained a Bachelor's degree in civil engineering Northeast Electric Power University in 2009.Currently, he is working as a senior engineer at Jiaxing Hengchuang Electric Power Design and Research Institute. His areas of interest is power transmission line design technology.

Lankang Zhu obtained a Bachelor's degree in Electrical Engineering and Automation Engineering from Nanjing University of Posts and Telecommunications in 2015.He obtained a Master's degree in Electrical Engineering from Hehai University in 2017. Currently, he is working as a an engineer at the power transmission room of Jiaxing Hengchuang Electric Power Design and Research Institute. His areas of interest is power transmission line design technology.

Wei Han obtained his master's degree in Civil Engineering from Northeast Electric Power University in 2015. Currently, he is engaged in the design work of power transmission projects at Jiaxing Hengchuang Electric Power Design and Research Institute. His areas of interest are the design, construction, and operation of high-voltage transmission lines.

Fa Wang obtained a Bachelor's degree in Transmission Line Engineering from Wuhan University of Water Resources and Electric Power in 2003. He obtained a Master's degree in Electrical Engineering from Tsinghua University in 2013. Currently, he is working as the president of Jiaxing Hengchuang electric power design and Research Institute. His areas of interest is emerging power technology.