Q-learning Guided Grey Wolf Optimizer for UAV 3D Path Planning

Binbin Tu[1], Fei Wang[2*], Xiaowei Han[3], Xibei Fu[4]
College of Intelligent System Science and Engineering, Shenyang University, Shenyang, China[1]
College of Information Engineering, Shenyang University, Shenyang, China[2, 4]
Institute of Innovation Science & Technology, Shenyang University, Shenyang, China[3]

***Abstract*—Path planning is a critical component of autonomous unmanned aerial vehicle (UAV) navigation systems, yet traditional and sampling-based methods encounter limitations in three-dimensional (3D) path planning. This paper offers a structured review of applicable algorithms in 3D space, introduces the state-of-the-art techniques, and addresses cutting-edge challenges associated with UAV heuristic decomposition methods. Furthermore, we develop a Q-learning guided grey wolf optimizer (QGWO) to tackle the UAV 3D path planning problem in complex scenarios. QGWO incorporates two exploration strategies from the aquila optimizer into the grey wolf optimizer, enhancing its capacity to escape local optima and utilize the population for broader exploration. Q-learning guides the search process, enabling the algorithm to store iterative information, accelerate convergence, and balance exploration and exploitation. Additionally, Laplace crossover perturbs the positions of the α and β wolves, preventing the algorithm from becoming trapped in local optima. To validate its effectiveness, QGWO and ten advanced heuristic algorithms were tested in 3D path planning simulations across six terrain scenarios of varying complexity. Experimental results demonstrate that QGWO achieves optimal cost metrics, outperforming the original grey wolf optimizer by up to 1.34% and significantly surpassing other algorithms with a 70.92% reduction in standard deviation. This highlights the effectiveness and robustness of QGWO in 3D path planning for UAV. Moreover, the Wilcoxon rank sum test shows that the null hypothesis is rejected in 98.33% of cases, confirming the statistical superiority of the proposed QGWO.**

Keywords*—*Q-learning; grey wolf optimizer; laplace crossover; 3D path planning; optimization

I. INTRODUCTION

Unmanned aerial vehicles (UAV) are extensively used in various military and civil tasks, including search and patrol, reconnaissance and surveillance, disaster rescue and relief, logistics, power inspection, and agricultural irrigation, owing to their high flexibility, mobility, low safety risk, and cost-effectiveness [2,18,22,52]. UAV path planning is a critical aspect of UAV technology. Initially, it focused on optimizing path length for simple point-to-point planning. However, it has evolved to consider complex conditions such as terrain, weather, threats, and obstacles, which collectively shape the UAV's flight environment. The goal is to find the safest, lowest-cost flight paths while considering the UAV's performance constraints and security threats.

The path planning problem is widely recognized as a challenging nonlinear NP-hard optimization dilemma, whose complexity escalates swiftly with problem size [4,6]. Over the past decades, researchers have devised a spectrum of methodologies to tackle this intricate optimization challenge. As depicted in Fig. 1, these methodologies typically fall within three main categories: traditional path planning algorithms, sampling-based path planning algorithms, and intelligent bionic algorithms. Notable examples include the A* algorithm [17, 31], the artificial potential field method [37], the Voronoi diagram algorithm [16], rapidly-exploring random trees [7,50], reinforcement learning-based path planning algorithms [47], as well as genetic algorithm [33,48,55], ant colony algorithm [19, 25], and particle swarm algorithm [32,35], among others. These sophisticated algorithms meticulously factor in not merely the pursuit of the shortest feasible path but also a myriad of pivotal considerations, encompassing obstacle evasion, energy expenditure optimization, and velocity management, among others. To adeptly navigate the complexities of diverse environmental conditions, they employ a plethora of strategic approaches and algorithmic refinements, notably hierarchical planning methodologies, dynamic map reconfiguration techniques, and intricate inter-layer connectivity strategies. This multifaceted approach ensures that the algorithms are adept at adapting to the unique flight requirements posed by varying environments, thereby enhancing their overall efficacy and robustness [22, 23].

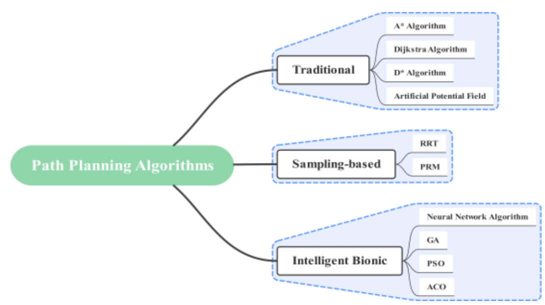


Fig. 1. Classification of path planning algorithms.

Generally, traditional path planning algorithms and sampling-based path planning algorithms are primarily suitable for straightforward spatial path planning tasks. However, when confronted with complex problems, their efficacy diminishes exponentially with increasing dimensionality, thereby presenting inherent limitations. Among the array of approaches, bionic group intelligence algorithms have garnered significant attention in recent years for addressing UAV path planning challenges due to their simplicity and effectiveness. Moreover, with the advancement of computer and big data technologies, the advantages of group intelligence algorithms, such as robustness, positive feedback mechanisms, and self-organization, have become increasingly pronounced.

In 3D environment path planning, the planner must possess enhanced capabilities to avoid terrain collisions and handle complex high-dimensional problems, surpassing the requirements of two-dimensional path planning. For instance, the grey wolf optimizer (GWO) [27] theoretically identifies the optimal path solution, making it suitable for path planning in intricate environments. The difficulty and uncertainty in UAV trajectory planning for complex environments escalate due to the intricate and variable nature of the terrain, which includes unknown slopes, gullies, and obstacles of various sizes and types. Primitive GWO often becomes trapped in local optima, particularly when handling highly complex and constrained tasks. Therefore, as terrain complexity and search space variability increase, it is crucial to design algorithms with greater population diversity, improved exploration and exploitation capabilities, and significant jumping ability within a limited response time. These enhancements will improve the stability and efficiency of trajectory planning for UAV in complex environments.

To advance research in this field and address the challenges of UAV trajectory planning in complex environments, this paper introduces a Q-learning guided grey wolf optimizer (QGWO). This novel approach provides a robust and highly accurate path planning method with superior performance. The paper is organized as follows: Section II reviews related work; Section III outlines the formulation of the objective function; Section IV details the proposed QGWO; Section V examines the 3D path planning capabilities of the QGWO across six different levels of complexity; and Section VI presents the conclusions.

II. Related Works

Recent improvements in the grey wolf optimizer for UAV 3D path planning can be categorized into three main areas: (1) enhancement of the initialization population, (2) optimization of parameters, and (3) refinement of the search mechanism. Aslan [3] et al. designed and introduced a greedy algorithm called back and forth to solve the path planning problem, where the heuristic is responsible for generating two antecedent paths and combining the generated paths in order to take advantage of their favorable line segments when obtaining a safer, shorter, and more maneuverable path candidate. Rao [34] et al. designed a dyadic-based learning model influenced by the refraction principle, together with an improved convergence factor to develop a multi-strategy collaborative grey wolf optimizer. Nadimi-Shahraki [28] et al. introduced a novel locomotor strategy known as the dimensional learning-based hunting search strategy, which draws inspiration from the natural hunting behaviors of wolves. Gupta [13] et al. developed an enhanced algorithm named RW-GWO, which is based on randomized wandering. Saremi [38] et al. incorporated evolutionary population dynamics into the grey wolf optimizer, thereby eliminating weaker search agents and repositioning them around the alpha, beta, or delta wolves to improve exploitation. Lu [21] et al. proposed the cellular grey wolf optimizer (CGWO) with a unique topological structure where each wolf has its own set of topological neighbors. This structure restricts interactions to neighboring wolves and employs an overlapping information dissemination mechanism to maintain population diversity over extended periods. Rodríguez [36] et al. introduced a hierarchical operator inspired by the hierarchical social structure of grey wolves, which models the hunting process within the algorithm, and proposed five variants.

Moreover, hybrid algorithms have increasingly become a focal point of improvement efforts, combining two or more algorithms to maximize their respective advantages. For instance, Teng [41] et al. utilized Tent chaotic sequences to initialize individual positions, introduced non-linear control parameters to balance the algorithm, and incorporated the particle swarm optimization concept to update each grey wolf's position based on both individual and pack optima. Najma [29] et al. exploited the exploration benefits of the whale optimization algorithm and the efficient exploitation of particle swarm optimization to specifically address the challenge of non-complete constraints in complex terrain. Wang [42] et al. hybridized the grey wolf optimizer with the Harris hawk optimization, endowing the grey wolf with the ability to "fly" and incorporating a nonlinear convergence factor with local perturbations and extended exploration strategies. Zhao [54] et al. introduced a convergence strategy for the golden sine optimizer to adapt the path planning problem for a cross-barrier robot by adjusting the working environment model, path generation method, and fitness function. Gaidhane [11] et al. enhanced exploration by integrating the information-sharing strategy from the artificial bee colony, while retaining the grey wolf optimizer's exploitation capabilities through its leadership hierarchy. Han [14] et al. proposed an IGWO-IAPF algorithm based on the fusion of an improved grey wolf optimizer and an improved artificial potential field algorithm. Waypoint attraction is added to the traditional artificial potential field algorithm based on force field, and an optimized individual position update strategy is used to coordinate the search capability of the algorithm to meet the requirements of 3D path planning. Wang [44] et al. introduced an eight-node search method into the traditional A-star algorithm to minimize the steering tendency of artificial intelligence transportation robots. Loganathan [20] et al. navigated efficiently and obstacle-avoidantly towards a target in less time by synergizing the strengths of two heuristic algorithms. Niu [30] et al. generated a real-time dynamic path planning method based on an improved interferometric hydrodynamic system and artificial neural networks to improve path quality and computational efficiency. This combination of improved algorithms results in relatively smooth and cost-effective UAV paths.

It is worth noting that certain complex strategies require the tuning of numerous parameters, and optimization problems are highly sensitive to these parameter settings [5,46]. Therefore, enhancing the convergence speed and solution quality of the planner, while ensuring that UAV motion is collision-free and feasible, remains a significant challenge. Although many researchers have extensively employed meta-heuristics to address UAV path planning and some have achieved notable improvements with evolutionary algorithms, the overall research results in UAV path planning remain limited. There is a need for more in-depth study of the two core mechanisms of population intelligence algorithms: exploitation and exploration.

III. PATH PLANNING COST FUNCTION

Assuming the UAV maintains a predetermined flight speed, the path planning problem can be simplified to a static path planning minimization problem. To align with the actual requirements of UAV operations, the cost function for UAV path planning is defined by considering the range length cost, flight altitude cost, threat cost, and smoothing cost. These factors are mathematically represented as follows.

$$F\left(X_i\right)=\sum_{k=1}^{4} b_k F_k\left(X_i\right) \tag{1}$$

Where X_i is the decision variable, a list of n waypoints (x, y, z); b_k is the weight of each cost function, set with reference to the literature [32]; and F_k is the kth cost function.

A. Cost of Track Length

The length of a UAV flight path should be as short as possible to save energy consumption. If the flight path is transformed into a number of waypoints to be flown over by the UAV, the Euclidean distance between two neighbouring waypoints is taken as the length of each segment, and the cost of the range length for a particular path is then calculated as follows.

$$F_1\left(X_i\right)=\sum_{i=1}^{n-1} \left\| \vec{P}_{ij}\vec{P}_{i,j+1} \right\| \tag{2}$$

B. Cost of Threat

Considering the complexity of threat modeling and the difficulty of obtaining real data, the threat environment is simplified by representing the threat region as a cylinder with a constant radius. The radius of action of the threat region is equal to the radius of the cylinder, as shown schematically in Fig. 2.

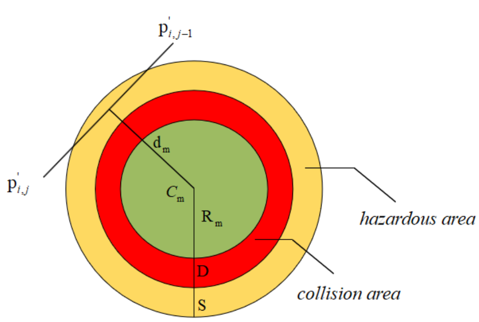


Fig. 2. Scope of the flight threat.

For a section of path, the threat cost is calculated as:

$$F_2(X_i)=\sum_{j=1}^{n-1}\sum_{m=1}^{M} T_m(\vec{P}_{ij}\vec{P}_{ij+1}) \tag{3}$$

$$T_m=\begin{cases} 0, if\, d_m > S+D+R_m \\ (S+D+R_m)-d_m, if\, D+R_m < d_m,, S+D+R_m \\ \infty, if\, d_m,, D+R_m \end{cases} \tag{4}$$

C. Cost of High

The UAV's flight altitude is typically constrained by both minimum and maximum altitude limits. The altitude cost is calculated as follows.

$$F_3(X_i)=\sum_{j=1}^{n} H_{ij} \tag{5}$$

$$H_{ij}=\begin{cases} \left| h_{ij}-\dfrac{(h_{\max}+h_{\min})}{2} \right|, if \quad h_{\min}\le h_{ij}\le h_{\max} \\ \infty, otherwise \end{cases} \tag{6}$$

D. Cost of Smoothing

The primary flight angle control parameters for a UAV are the horizontal steering angle and the vertical pitch angle. These parameters must comply with the UAV's actual angle constraints; otherwise, the trajectory planning model will fail to produce a viable flight path. The horizontal steering angle φ_{ij} is the angle between two consecutive path segments projected onto the horizontal plane, and the vertical pitch angle θ_{ij} is the angle between two consecutive path segments projected onto the vertical axis. These angles are calculated as follows.

$$\varphi_{ij}=\arctan\left(\frac{\left\| \vec{P}'_{ij}\vec{P}'_{i,j+1} \times \vec{P}'_{i,j+1}\vec{P}'_{i,j+2} \right\|}{\vec{P}'_{ij}\vec{P}'_{ij+1}\cdot \vec{P}'_{i,j+1}\vec{P}'_{ij+2}}\right) \tag{7}$$

$$\theta_{ij}=\arctan\left(\frac{z_{i,j+1}-z_{ij}}{\left\| \vec{P}'_{ij}\vec{P}'_{i,j+1} \right\|}\right) \tag{8}$$

$$F_4(X_i)=a_1\sum_{j=1}^{n-2}\varphi_{ij}+a_2\sum_{j=1}^{n-1}\left|\theta_{ij}-\theta_{i,j-1}\right| \tag{9}$$

IV. PROPOSED QGWO

A. Grey Wolf Optimizer

The implementation of the grey wolf optimizer involves four classes of grey wolves: α, β, δ, and ω. In the optimization process, each wolf's position represents a potential solution. The global solution is identified by using the α, β, and δ wolves as the optimal, suboptimal, and better solutions, respectively, while directing the ω wolves to promising regions. The grey wolf pack progressively approaches and encircles the prey as described by Eq. (11).

$$D_r=\left|C\cdot X_p(t)-X(t)\right| \tag{10}$$

$$X(t+1) = X_p(t) - A \cdot D \tag{11}$$

Where t is the current iteration number, $X_p(t)$ is the position of the prey; D_r is the encircling step; A and C are random vectors, determined by Eq. (12) and Eq. (13).

$$A = 2a \cdot r_1 - a \tag{12}$$

$$C = 2 \cdot r_2 \tag{13}$$

Where r_1 and r_2 are random numbers between [0,1]; a is the convergence factor, which decreases linearly from 2 to 0 as the number of iterations increases.

$$\begin{cases} X_1(t+1) = X_\alpha(t) - A \cdot \left| C \cdot X_\alpha(t) - X(t) \right| \\ X_2(t+1) = X_\beta(t) - A \cdot \left| C \cdot X_\beta(t) - X(t) \right| \\ X_3(t+1) = X_\delta(t) - A \cdot \left| C \cdot X_\delta(t) - X(t) \right| \end{cases} \tag{14}$$

$$X(t+1) = (X_1 + X_2 + X_3)/3 \tag{15}$$

The other grey wolves in the population update their positions based on the positions of the α, β, and δ wolves.

B. Position Update Incorporating the Aquila Optimizer

The grey wolf optimizer, with its simple structure, strong local search capability, and ease of application, has a certain degree of competitiveness among intelligent algorithms. However, under the influence of the linear convergence factor, search mechanism and single position update method, it is difficult to balance local exploration and global exploitation, i.e., the phase transition of the grey wolf optimizer is completely determined by a.

Thus, the hybrid algorithm is enhanced by integrating the extended and reduced exploration strategies from the aquila optimizer (AO) [1], known for its robust global search capability. The primary objective is to augment the algorithm's capacity to escape local optima while leveraging the population to explore the solution space comprehensively.

$$X(t+1) = X_\alpha(t) \cdot (1 - t/T) + (X_M(t) - X_\alpha(t) \cdot r_1) \tag{16}$$

$$X_M(t) = \frac{1}{N} \sum_{i-1}^{N} X(t) \tag{17}$$

In the extended exploration strategy, the aquila optimizer completes the identification of the prey area and conducts high-altitude flight followed by a vertical dive to select the optimal hunting area. During this process, the aquila flies at a high altitude to pinpoint the prey location, characterized by its mathematical model shown in Eq. (16), where $X_M(t)$ represents the average position of all individuals and N denotes the population size.

$$X(t+1) = X_\alpha(t) \cdot levy(D) + X_R(t) + (y - x) \cdot r_2 \tag{18}$$

$$\sigma = \Gamma(1+\beta) \cdot \sin(\pi\beta/2) \Big/ \Gamma(1+\beta/2) \cdot \beta \cdot 2^{\beta-1/2} \tag{19}$$

$$levy(D) = 1.5 \cdot u \cdot \sigma \Big/ |v|^{\frac{1}{\beta}} \tag{20}$$

$$x = (r_3 + U \cdot D_1) \cdot \sin(-0.005 \cdot d + 3\pi/2) \tag{21}$$

$$y = (r_3 + U \cdot D_1) \cdot \cos(-0.005 \cdot d + 3\pi/2) \tag{22}$$

The reduced exploration strategy corresponds to the isometric flight of the aquila's short gliding attack, during which the aquila locks the prey target from high altitude and hovers above the target to prepare for launching the attack, and its mathematical model is shown in Eq. (18). Where, $levy(D)$ is the Lévy flight distribution number and D is the dimension of the problem; $X_R(t)$ is the position of the random individual at t iterations; x and y denote the random shapes in the search; $r_3 \in [1,20]$ is used to fix the search period; U is a fixed constant and d is an integer within [1, D].

Compared to the original grey wolf optimizer, the position update after incorporating the aquila optimizer makes full use of the available search information, incorporates interactions between individuals, and weakens the absolute influence of α, β, and δ wolves on individuals. The average position of all individuals is considered when performing the position update in the expanded exploration, instead of simply moving only in the direction of the three wolves with the lowest fitness values. Furthermore, the Lévy flight used in the reduced exploration avoids overdependence of the grey wolf population to fall into a local optimum due to the fact that α wolves have much lower fitness values than β and δ wolves, and the inclusion of the position of random individuals weakens the influence of the α wolves, while at the same time increasing the randomness of the algorithm.

C. Q-learning Guided Search

The grey wolf optimizer inherently lacks the capability to store search information, relying solely on fitness comparisons to identify the top three ranks of grey wolves. This, coupled with an uneven exploration and exploitation phase, poses challenges in escaping local optima. To address this limitation, Q-learning is introduced to guide the search of the grey wolf optimizer with an improved position updating method. In Q-learning, the Q-table can be regarded as the agent's experience, while the reward table represents a combination of behaviors and states, rewarding or punishing the agent accordingly [51]. Each agent accumulates experience and selects optimal actions by exploring the environment in specific iterations, updating the corresponding Q-value according to the Bellman equation.

$$\begin{aligned} Q(s^{t+1}, a^{t+1}) &= Q(s^t, a^t) + \\ &\lambda[R_{t+1} + \gamma \max_a Q(s^{t+1}, a^t) - Q(s^t, a^t)] \end{aligned} \tag{23}$$

Where λ is the system learning rate and $0<\lambda\leq 1$; γ is the reward decay rate and $0<\gamma\leq 1$; R is the reward value; and $\max_a Q(s^{t+1},a^t)$ denotes the gain of the highest value action in the next state.

In the improved algorithm, where the grey wolf population acts as an agent and the choice of location update methods acts as a collection of agent actions, then a mapping between the improved grey wolf optimizer and Q-learning can be achieved. Q-learning helps the algorithm to store information about the search space gained during iterations, giving positive rewards to well-performing prey and negative rewards to poorly-performing prey, and helps to find from the three location update methods the most suitable choice, making the exploration and development phases of the algorithm more balanced. The primary Q-learning-based guided search can be summarized in the following five steps.

1) Initialise the Q-table as the zero matrix with the following reward table;

$$\begin{bmatrix} -1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{24}$$

2) Determine the current best action for the iteration based on the values stored in the current state Q-table;

3) Execute the selected action and compute the new adaptation and reward, calculated as follows;

$$reward = \begin{cases} 1 \quad if \quad fitness \;\; increases \\ -1 \;\; if \quad fitness \;\; decreases \end{cases} \tag{25}$$

4) Use (23) to update the Q-table;

5) Determine the iteration termination condition of the algorithm and repeat the execution if it is not satisfied.

D. Laplace Crossover Variant Perturbation

In the later iterations of the original grey wolf optimizer, individual grey wolves tend to rapidly converge and cluster near the current optimal position. If this position is not the global optimum, the grey wolf population may struggle to explore beyond the limited search range, thus failing to discover the global optimal position and becoming trapped in a local optimum. Introducing mutation perturbation can enhance the diversity of the grey wolf population [12], enabling the algorithm to escape local optima and explore other regions of the solution space. This increases the likelihood of finding the global optimum.

Common variation perturbations include the introduction of differential operators, Cauchy operators [45], Gaussian operators [53], etc. Differential operators involve mutation and selection operations, whereas Cauchy operators and Gaussian operators are commonly used for all individuals throughout the iterative process, which leads to an increase in algorithm complexity. For this reason, the Laplace cross mutation is used to perturb the positions of α and β wolves after updating the original grey wolf optimizer, and the fitness is calculated to select the best. The Laplace distribution density function and the α and β wolf position cross variations are as follows.

$$f(x)=\frac{1}{2b}e^{-\frac{|x-\mu|}{b}} \tag{26}$$

$$X_\alpha(t+1)' = X_\alpha(t)+\beta\left|X_\alpha(t)-X_\beta(t)\right| \tag{27}$$

$$X_\beta(t+1)' = X_\beta(t)+\beta\left|X_\alpha(t)-X_\beta(t)\right| \tag{28}$$

Where μ is a position parameter and a real number, b is a scale parameter and greater than 0, and β is the Laplace random distribution number. Shail [9] et al. introduced the Laplace operator after each iteration for the variance perturbation, without considering the effect of the perturbation amplitude on the results in different iteration periods. From the Laplace density function curves under different b values in Fig. 3, it can be seen that, from the vertical direction, the peak near the center of $b=0.5$ is larger than that of $b=1$, while the peaks at the two ends are smaller, and the probability of generating random numbers in the center region is higher. From the horizontal direction, the closer to the ends of the horizontal axis at $b=1$ the slower the decline, and the easier it is to generate random numbers away from the origin.

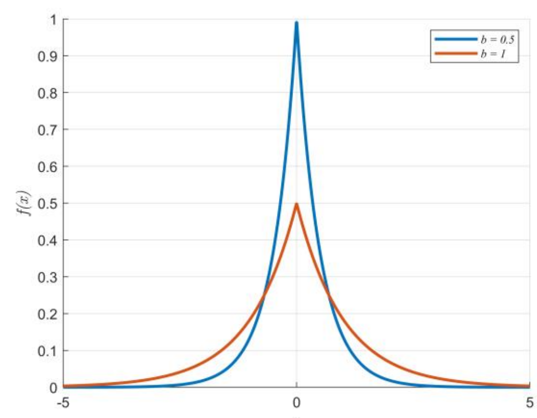


Fig. 3. Laplace density functional curves.

To efficiently escape local optima, the Laplace distribution density function with a scale parameter of $b=0.5$ is employed to perturb the positions of the α and β wolves in the later iterations of the original grey wolf optimizer. This choice is motivated by the higher probability of generating random numbers near the central value, which enables the grey wolf to finely search the optimal region with a smaller step size. Consequently, it increases the likelihood of discovering the global optimal solution.

E. QGWO Algorithm Flow

QGWO distinguishes itself through its capacity to store search information throughout the iteration process and dynamically switch between various position updating methods

to efficiently locate the global optimal solution. Drawing from the exposition in the initial four subsections of this chapter, the algorithmic flowchart of QGWO is depicted in Fig. 4.

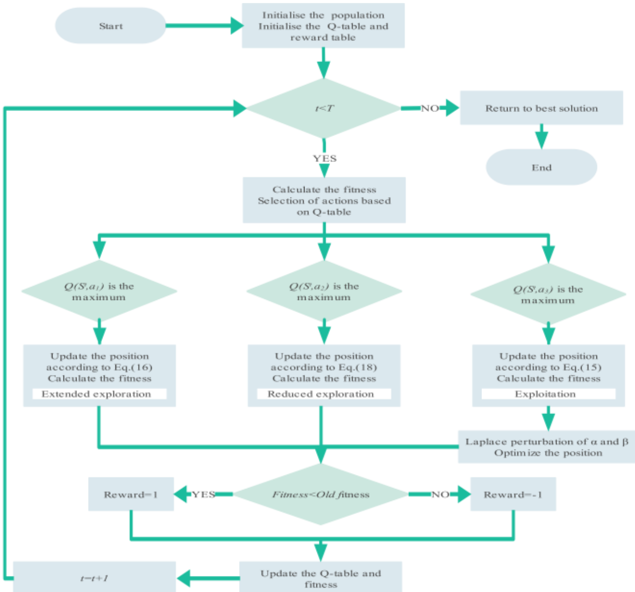


Fig. 4. Flowchart of QGWO.

QGWO distinguishes itself through its capacity to store search information throughout the iteration process and dynamically switch between various position updating methods to efficiently locate the global optimal solution. Drawing from the exposition in the initial four subsections of this chapter, the algorithmic flowchart of QGWO is depicted in Fig. 4.

V. Simulation Experiments and Analyses

The main objective of this section is to provide an in-depth study of the performance of the newly proposed QGWO. To ensure the comprehensiveness of the analysis, we apply QGWO to six 3D path planning models of different complexity, complemented by 10 algorithms for comparison. The algorithms used for comparison include the standard GWO, three excellent improved versions: AGWO [24], AGWO-CS [39], PSOGWO [40], and the widely cited and excellent algorithms PSO [43], WOA [26], HHO [15], SOA [8], MPA [10], DBO [49]. It is noteworthy that the specific parameter settings of all algorithms utilized for comparison were directly sourced from the references without any alterations. This rigorous benchmarking approach establishes a robust foundation for evaluating the optimization capabilities of QGWO.

A. Simulation of Terrain Environment and Parameters

To effectively validate the efficacy of the improved algorithm, a real digital elevation model of Christmas Island, Australia, along with another data map featuring diverse terrain structures, was utilized as the terrain environments in the UAV path planning problem. Various threat objects, depicted as black cylinders, were introduced to simulate terrains of varying complexity, resulting in a total of six terrain scenarios for path planning comparison tests. The specific map ranges and start/end coordinates are outlined in Table I. Each algorithm employed in the simulation experiments was configured with a maximum iteration number of 500 and a population size of 30. Furthermore, each algorithm was run independently for 30 iterations to mitigate any chance factors.

TABLE I. Map Information

Scenario	*Map range*	*Start*	*End*
1,2,3	[1100,900]	[200,100,150]	[800,800,150]
4,5,6	[450,450]	[10,10,200]	[400,400,150]

B. Compared to Other GWO Variants

GWO, AGWO, AGWO-CS, PSOGWO, and QGWO are combined for trajectory planning based on the above environments, and the cost curves and top views of the planning results are shown in Fig. 5 and Fig. 6. Among the six scenarios featuring diverse terrains and threats, the data presented in Table II demonstrates the significant superiority of QGWO in terms of optimal path planning cost. Compared to GWO and other variants, QGWO achieves the best cost optimal cost every time and effectively reduces it by 1.34%, reduces the standard deviation by 70.92%, and has the smallest average of 30 independent runs, inferior to AGWO-CS only in terms of standard deviation in scenario 2. The data depicted in Table II illustrates the effectiveness of the QGWO enhancement, evidencing its competitive edge. Furthermore, the methodology sustains a balanced approach between developmental and exploratory phases, which guarantees consistent performance amidst diverse complexity scenarios. Such stability is pivotal for real-world deployments, bolstering the method's credibility and reliability in practical applications.

TABLE II. Path Planning Costs Generated By QGWO and Other GWO Variants

	GWO			***AGWO***			***AGWO_CS***			***PSOGWO***			***QGWO***		
Scenario	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
1	9290.96	9537.12	320.56	9297.86	9535.24	286.87	9299.51	9345.32	49.53	9271.55	10330.33	1364.60	**9265.42**	**9309.16**	**18.26**
2	9335.27	10258.89	884.43	9474.24	10741.11	508.29	9392.15	9622.91	**126.71**	9403.97	11561.63	1235.35	**9299.06**	**9550.25**	266.85
3	9950.42	10170.02	526.95	10041.90	11507.14	710.08	10080.10	11046.29	1035.89	10011.20	13025.64	2097.27	**9891.56**	**9988.09**	**47.31**
4	5908.81	7010.61	870.20	6105.56	7271.93	836.50	6222.64	6704.75	438.19	5834.98	7396.41	1311.54	**5829.74**	**6366.13**	**422.13**
5	5890.74	6601.78	785.19	6122.26	7633.97	1009.59	6116.76	6621.61	508.68	5877.68	7570.84	1244.54	**5869.42**	**6188.52**	**303.27**
6	5985.94	6681.50	823.65	6264.61	7622.10	993.67	6045.12	6541.61	564.69	5946.54	7536.58	1308.84	**5944.27**	**6222.63**	**350.02**

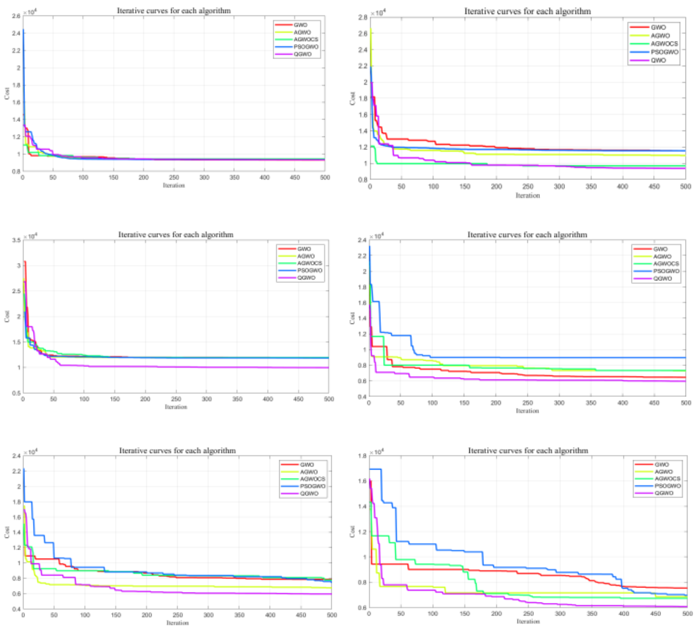
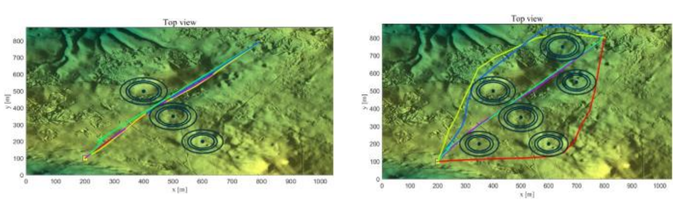


Fig. 5. Convergence curves of QGWO and other GWO variants.

Fig. 5 provides an in-depth analysis of the algorithm's convergence behavior by illustrating the evolution of the best fitness across iterations for each variant. In simpler scenarios, QGWO demonstrates significantly faster convergence compared to most GWO variants. Additionally, QGWO effectively navigates local optimum traps. In the most complex scenario (scenario 6), the flight paths generated by GWO tend to be longer, resulting in increased fuel consumption. More critically, its flight paths are closer to risky regions, posing a major safety hazard and increasing the likelihood of flight accidents. Q-learning enhances QGWO by storing information about the search space throughout the iterations. This allows the algorithm to select an optimal position update method for each iteration. Consequently, individuals can escape local optima through a versatile position update formula and a variant perturbation strategy.



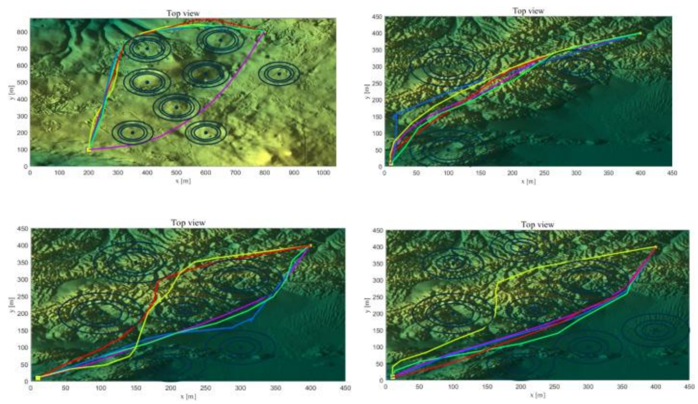


Fig. 6. Top view of QGWO and other algorithms.

The top view of Fig. 6 specifically shows the paths planned by each algorithm, and it can be directly seen that the paths generated by GGWO in scenes 2, 3, and 4 are too curved and are not optimal, i.e., there is a deficiency in the global search capability of 3D path planning. However, QGWO performs well in different types of scenarios, and even in scenario 3 only QGWO finds the globally optimal path.

C. Compared to Other Heuristic Algorithms

To effectively showcase the outstanding performance of QGWO in UAV 3D path planning, the six aforementioned scenarios are utilized to compare QGWO with several other competitive state-of-the-art algorithms.

According to the data in Table III, DBO achieves the optimal path planning cost in scenario 3, with QGWO ranking second, trailing by only 6.23 units. However, in the remaining five scenarios, QGWO ranks first across all three metrics, demonstrating its reliability. The convergence curves in Fig. 7 show that QGWO has the lowest initial best fitness value and the fastest convergence rate. It converges more rapidly in the complex scenarios 3, 4, 5, and 6, particularly in the later iterations when most algorithms fall into local optima. QGWO relies on the Laplace variant to explore the current position in small steps, allowing it to continue converging. This indicates that QGWO not only starts from a better position but also reaches the target solution more efficiently than the control algorithms, highlighting its superior optimization capability. Overall, QGWO, enhanced by integrating multiple strategies, performs exceptionally well in UAV 3D path planning, further validating the effectiveness of the proposed enhancement strategies.

TABLE III. PATH PLANNING COSTS GENERATED BY QGWO AND OTHER HEURISTIC ALGORITHMS

	PSO			***WOA***			***HHO***			***SOA***			***DBO***			***MPA***			***QGWO***		
Scenario	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
1	9580.81	11071.29	905.76	9281.60	9461.02	577.34	9326.95	9708.57	624.63	9288.15	9489.44	205.41	9270.45	10818.66	1319.26	9317.93	10460.94	763.54	**9265.42**	**9309.16**	**18.26**
2	9911.85	11700.20	1198.47	9543.42	9837.01	758.51	9548.51	9917.57	648.29	9616.01	10020.33	480.80	9541.95	12286.55	1685.02	9557.71	11623.43	1022.55	**9299.06**	**9550.25**	**266.85**
3	10591.39	12905.73	1575.	10050.51	1240	1816.	1273	14919.2	871.43	10368.8	12428.0	1061.7	**9885.3**	12966.7	1700.0	10510.8	13162.3	998.25	9891.5	**9988.0**	**47.31**

	PSO			WOA			HHO			SOA			DBO			MPA			QGWO		
			31		0.93	71	2.87	6		4	8	9	**3**	7	8	7	7		6	**9**	
4	6606.40	7864.94	744.14	6027.40	7958.71	1089.00	7186.74	10279.32	2285.95	7678.06	10215.67	1512.57	6001.02	8224.27	1932.50	6519.42	8163.36	817.13	**5829.74**	**6366.13**	**422.13**
5	6272.18	7671.81	1020.25	6311.68	8274.48	1495.03	7773.42	11136.27	2184.67	6787.47	10822.42	1757.76	5880.11	7701.94	1177.27	6153.88	7887.54	1057.43	**5869.42**	**6188.52**	**303.27**
6	6541.41	7700.60	861.11	6393.42	7818.07	1411.35	7337.43	10474.85	2259.84	7456.82	9721.14	1343.02	5952.18	7603.87	1112.63	6297.14	7873.92	979.33	**5944.27**	**6222.63**	**350.02**

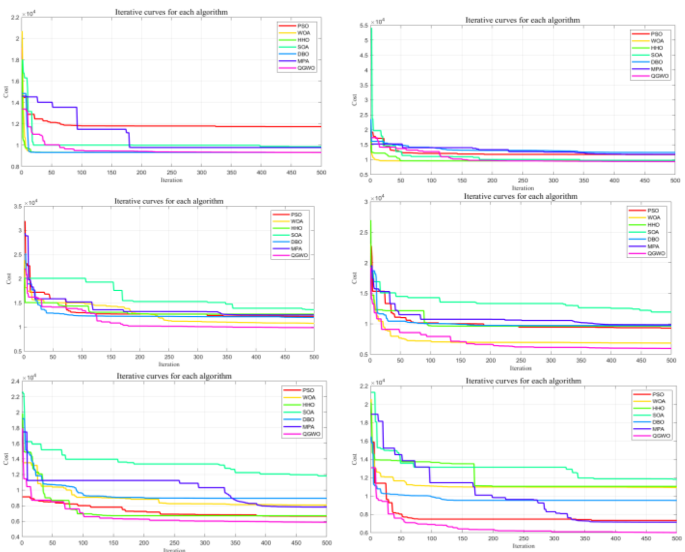
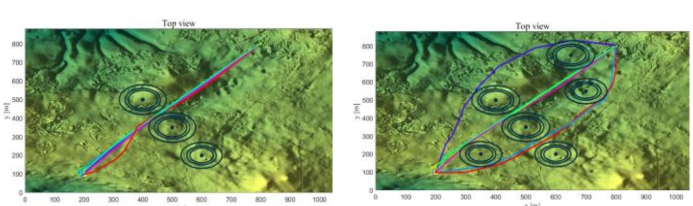


Fig. 7. Convergence curves of QGWO and other algorithms.

Fig. 8 shows the top view of the paths planned by the algorithms, illustrating their ability to safely navigate to the target point without colliding with obstacles. In the simplest scenario, there is minimal disparity between the paths planned by the various algorithms, noticeable only in the final cost. However, as the scenario's complexity increases and obstacles become denser, the cost disparity widens, highlighting the performance advantage of QGWO. In the remaining five scenarios, the algorithm labeled SOA performs the worst, often following the outermost obstacles and producing tangled routes. The paths planned by MPA and DBO successfully avoid risk areas but are uneven and feature sharp turns, increasing the flight distance. In contrast, the optimized QGWO effectively addresses these issues, resulting in smoother and more efficient paths. Paths generated by other comparison algorithms also exhibit more pronounced twists and turns.



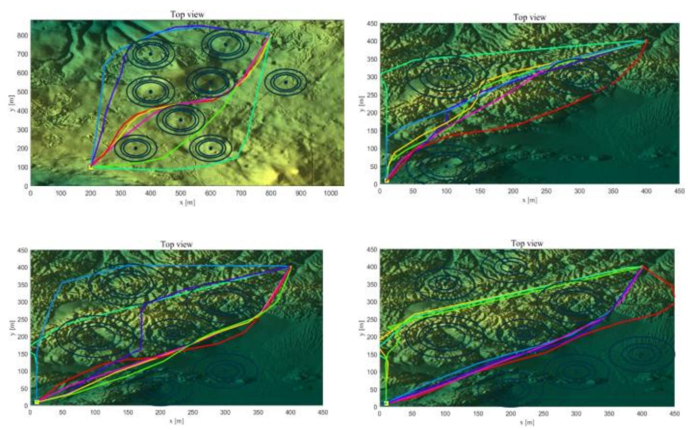


Fig. 8. Top view of QGWO and other algorithms.

Fig. 9 displays the side and 3D views of the paths generated by QGWO for scenarios 3 and 6, which are the two most complex scenarios. The depicted paths exhibit notable smoothness and efficiency, maintaining appropriate flight altitudes relative to the terrain. These visualizations underscore the QGWO's capability to navigate challenging environments effectively, avoiding obstacles while optimizing the flight trajectory. The algorithm's ability to generate such high-quality paths in the most complex scenarios highlights its robustness and reliability, further validating its suitability for real-world applications where terrain complexity poses significant challenges.

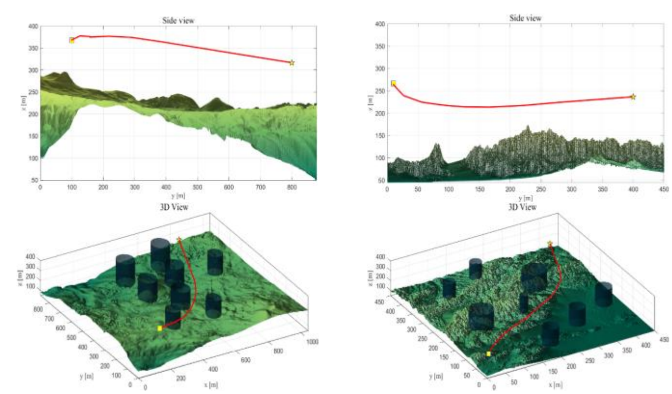


Fig. 9. QGWO path planning in scenarios 3 and 6.

D. Statistical Test

To compare the performance of the algorithms, the optimum, mean, and standard deviation were calculated from 30 experiments for each algorithm in the simulation. To further assess the differences between QGWO and other algorithms, a statistical analysis was conducted using the Wilcoxon rank-sum test with a 5% significance level. This test determined whether there were significant differences between QGWO and the other algorithms. The results are presented in Table IV.

TABLE IV. PATH PLANNING COSTS GENERATED BY QGWO AND OTHER HEURISTIC ALGORITHMS

Scenario	*PSO*	*WOA*	*HHO*	*SOA*	*DBO*	*MPA*	*GWO*	*AGWO*	*AGWO_CS*	*PSOGWO*
1	**2.92E-10**	**4.29E-08**	**2.26E-08**	**1.85E-07**	**5.35E-08**	**8.01E-12**	**6.14E-05**	**1.73E-06**	**4.74E-06**	**1.53E-05**
2	**3.02E-11**	**3.69E-11**	**3.34E-11**	**3.02E-11**	3.69E-01	**2.92E-12**	**2.31E-06**	**1.61E-10**	**2.38E-07**	**6.70E-11**
3	**3.25E-12**	**1.09E-10**	**3.02E-11**	**4.50E-11**	**1.17E-09**	**3.69E-11**	**1.60E-03**	**4.98E-11**	**5.10E-11**	**7.50E-11**
4	**2.09E-10**	**4.18E-09**	**5.49E-11**	**3.53E-10**	**1.93E-05**	**4.17E-11**	**3.42E-04**	**2.01E-07**	**4.46E-04**	**6.55E-04**
5	**2.23E-09**	**9.76E-11**	**3.02E-11**	**5.18E-10**	**7.75E-06**	**2.44E-09**	**3.08E-03**	**7.96E-08**	**4.80E-06**	**5.14E-06**
6	**2.15E-10**	**2.87E-120**	**3.02E-11**	**3.02E-11**	**1.29E-06**	**7.38E-10**	**6.97E-03**	**2.03E-09**	**2.49E-05**	**3.19E-07**

The test results in Table IV show that in 98.33% of cases, the p-value for comparisons between QGWO and other algorithms across six different complexity scenarios is less than 0.05, leading to the rejection of the null hypothesis. This indicates a significant difference in computational performance between QGWO and the other 10 algorithms, confirming the statistical superiority of the proposed QGWO.

VI. CONCLUSION

This paper introduces an enhanced version of the Q-learning guided optimization algorithm, termed QGWO, to address certain limitations of the grey wolf optimizer in UAV 3D path planning applications. QGWO acknowledges the stringent constraints inherent in UAV 3D path planning and, firstly, integrates two robust global search strategies from the aquila optimizer into the original grey wolf population update process, thereby enhancing the algorithm's capability to escape local optima. Secondly, recognizing that the phase transition of the original grey wolf optimizer is solely determined by the convergence factor, Q-learning is introduced to guide the search process after improving the position update method. This facilitates the algorithm in storing iterative information, accelerates population convergence, and balances exploration and exploitation. Finally, Laplace cross variation is employed to perturb the positions of α and β wolves post-updating the original grey wolf optimizer, aiding in fine-tuning the search within the optimal region and thereby improving the likelihood of finding the global optimal solution.

For the 3D path planning problem, the simulation experiments design a cost function with multi-factor co-constraints and construct six terrain scenarios of varying complexity. Comparison experiments between QGWO and ten other high-performing heuristic algorithms conclude that QGWO exhibits exceptional performance in both acquiring optimal cost and delivering high-quality, stable solutions.

Future research will aim to fully realize the potential of QGWO. We are particularly interested in exploring its application in various domains such as logistics, healthcare, and energy management, where its optimization capabilities can address complex challenges. Additionally, we plan to enhance the algorithmic structure of QGWO to further improve its performance and investigate its integration with other computational techniques, such as machine learning models, to create hybrid approaches. This future work aims not only to push the boundaries of algorithmic optimization but also to provide practical solutions to pressing problems across various industries.

ACKNOWLEDGMENT

This research was funded by the National Natural Science Foundation of China: Grant No. 71672117; Liaoning Provincial Science and Technology Programme: Grant No. 2023JH2/101300205; Shenyang Science and Technology Programme: Grant No. 23-407-3-33; Liaoning Province Graduate Education and Teaching Reform Research Project：Grant No. LNYJG2024330.

REFERENCES

[1] L. Abualigah, D. Yousri, M. Abd Elaziz, A.A. Ewees, M.A.A. Al-qaness, A.H. Gandomi, Aquila Optimizer: A novel meta-heuristic optimization algorithm, Comput. Ind. Eng. 157 (2021) 107250.

[2] B. Alzahrani, O.S. Oubbati, A. Barnawi, M. Atiquzzaman, D. Alghazzawi, UAV assistance paradigm: State-of-the-art in applications and challenges, J. Netw. Comput. Appl. 166 (2020) 102706.

[3] S. Aslan, Back-and-Forth (BaF): a new greedy algorithm for geometric path planning of unmanned aerial vehicles, Computing 106 (2024) 2811–2849.

[4] E. Besada-Portas, L. de la Torre, J.M. de la Cruz, B. de Andrés-Toro, Evolutionary Trajectory Planner for Multiple UAVs in Realistic Scenarios, IEEE Trans. Robot. 26 (2010) 619–634.

[5] F. Chen, N. Chen, H. Mao, H. Hu, Assessing four Neural Networks on Handwritten Digit Recognition Dataset (MNIST), (2019).

[6] F. Chen, Z. Luo, Y. Xu, D. Ke, Complementary Fusion of Multi-Features and Multi-Modalities in Sentiment Analysis, (2019).

[7] X. Cui, C. Wang, Y. Xiong, L. Mei, S. Wu, More Quickly-RRT*: Improved Quick Rapidly-exploring Random Tree Star algorithm based on optimized sampling point with better initial solution and convergence rate, Eng. Appl. Artif. Intell. 133 (2024) 108246.

[8] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, Knowl.-Based Syst. 165 (2019) 169–196.

[9] S.K. Dinkar, K. Deep, Opposition based Laplacian Ant Lion Optimizer, J. Comput. Sci. 23 (2017) 71–90.

[10] A. Faramarzi, M. Heidarinejad, S. Mirjalili, A.H. Gandomi, Marine Predators Algorithm: A nature-inspired metaheuristic, Expert Syst. Appl. 152 (2020) 113377.

[11] P.J. Gaidhane, M.J. Nigam, A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems, J. Comput. Sci. 27 (2018) 284–302.

[12] V. Garg, K. Deep, S. Bansal, Improved Teaching Learning Algorithm with Laplacian operator for solving nonlinear engineering optimization problems, Eng. Appl. Artif. Intell. 124 (2023) 106549.

[13] S. Gupta, K. Deep, A novel Random Walk Grey Wolf Optimizer, Swarm Evol. Comput. 44 (2019) 101–112.

[14] D. Han, Q. Yu, H. Jiang, Y. Chen, X. Zhu, L. Wang, Three-Dimensional Path Planning for Post-Disaster Rescue UAV by Integrating Improved Grey Wolf Optimizer and Artificial Potential Field Method, Appl. Sci. 14 (2024) 4461.

[15] A.A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, Future Gener. Comput. Syst. 97 (2019) 849–872.

[16] C. Huang, B. Du, M. Chen, Multi-UAV Cooperative Online Searching Based on Voronoi Diagrams, IEEE Trans. Aerosp. Electron. Syst. (2024) 1–12.

[17] R. Kala, A. Shukla, R. Tiwari, Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning, Artif. Intell. Rev. 33 (2010) 307–327.

[18] A.A. Khuwaja, Y. Chen, N. Zhao, M.-S. Alouini, P. Dobbins, A Survey of Channel Modeling for UAV Communications, IEEE Commun. Surv. Tutor. 20 (2018) 2804–2821.

[19] J. Liu, J. Yang, H. Liu, X. Tian, M. Gao, An improved ant colony algorithm for robot path planning, Soft Comput. 21 (2017) 5829–5839.

[20] A. Loganathan, N.S. Ahmad, A Hybrid HHO-AVOA for Path Planning of a Differential Wheeled Mobile Robot in Static and Dynamic Environments, IEEE Access 12 (2024) 25967–25979.

[21] C. Lu, L. Gao, J. Yi, Grey wolf optimizer with cellular topological structure, Expert Syst. Appl. 107 (2018) 89–114.

[22] Z. Luo, H. Xu, F. Chen, Audio Sentiment Analysis by Heterogeneous Signal Features Learned from Utterance-Based Parallel Neural Network, EasyChair, 2018.

[23] Z. Luo, X. Zeng, Z. Bao, M. Xu, Deep Learning-Based Strategy For Macromolecules Classification with Imbalanced Data from Cellular Electron Cryotomography, in: 2019 Int. Jt. Conf. Neural Netw. IJCNN, 2019: pp. 1–8.

[24] C. Ma, H. Huang, Q. Fan, J. Wei, Y. Du, W. Gao, Grey wolf optimizer based on Aquila exploration method, Expert Syst. Appl. 205 (2022) 117629.

[25] C. Miao, G. Chen, C. Yan, Y. Wu, Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, Comput. Ind. Eng. 156 (2021) 107230.

[26] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.

[27] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.

[28] M.H. Nadimi-Shahraki, S. Taghian, S. Mirjalili, An improved grey wolf optimizer for solving engineering problems, Expert Syst. Appl. 166 (2021) 113917.

[29] H.T. Najm, N.S. Ahmad, A.S. Al-Araji, Enhanced path planning algorithm via hybrid WOA-PSO for differential wheeled mobile robots, Syst. Sci. Control Eng. (2024).

[30] Y. Niu, X. Yan, Y. Wang, Y. Niu, 3D real-time dynamic path planning for UAV based on improved interfered fluid dynamical system and artificial neural network, Adv. Eng. Inform. 59 (2024) 102306.

[31] S.M. Persson, I. Sharf, Sampling-based A* algorithm for robot path-planning, Int. J. Robot. Res. 33 (2014) 1683–1708.

[32] M.D. Phung, Q.P. Ha, Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization, Appl. Soft Comput. 107 (2021) 107376.

[33] L. Ran, S. Ran, C. Meng, Green city logistics path planning and design based on genetic algorithm, PeerJ Comput. Sci. 9 (2023) e1347.

[34] C. Rao, Z. Wang, P. Shao, A Multi-Strategy Collaborative Grey Wolf Optimization Algorithm for UAV Path Planning, Electronics 13 (2024) 2532.

[35] V. Roberge, M. Tarbouchi, G. Labonte, Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning, IEEE Trans. Ind. Inform. 9 (2013) 132–141.

[36] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C.I. Gonzalez, G.E. Martinez, J. Soto, A fuzzy hierarchical operator in the grey wolf optimizer algorithm, Appl. Soft Comput. 57 (2017) 315–328.

[37] S.M.H. Rostami, A.K. Sangaiah, ** Wang, **aozhu Liu, Obstacle avoidance of mobile robots using modified artificial potential field algorithm, EURASIP J. Wirel. Commun. Netw. 2019 (2019) 1–19.

[38] S. Saremi, S.Z. Mirjalili, S.M. Mirjalili, Evolutionary population dynamics and grey wolf optimizer, Neural Comput. Appl. 26 (2015) 1257–1263.

[39] S. Sharma, R. Kapoor, S. Dhiman, A novel hybrid metaheuristic based on augmented grey wolf optimizer and cuckoo search for global optimization, in: IEEE, 2021: pp. 376–381.

[40] N. Singh, S. Singh, Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance, J. Appl. Math. 2017 (2017).

[41] Z. Teng, **-Ling Lv, L. Guo, An improved hybrid grey wolf optimization algorithm, Soft Comput. 23 (2019) 6617–6631.

[42] B. Tu, F. Wang, Y. Huo, X. Wang, A hybrid algorithm of grey wolf optimizer and harris hawks optimization for solving global optimization problems with improved convergence performance, Sci. Rep. 13 (2023) 22909.

[43] D. Wang, D. Tan, L. Liu, Particle swarm optimization algorithm: an overview, Soft Comput. 22 (2018) 387–408.

[44] J. Wang, N. Zhao, C. Mi, ART path planning method based on the 3D steering angle weighted A-star algorithm, Int. J. Veh. Inf. Commun. Syst. 9 (2024) 256–275.

[45] M. Wang, J.-S. Wang, X.-D. Li, M. Zhang, W.-K. Hao, Harris Hawk Optimization Algorithm Based on Cauchy Distribution Inverse Cumulative Function and Tangent Flight Operator, Appl. Intell. 52 (2022) 10999–11026.

[46] X. Wang, H. Zhao, T. Han, H. Zhou, C. Li, A grey wolf optimizer using Gaussian estimation of distribution and its application in the multi-UAV multi-target urban tracking problem, Appl. Soft Comput. 78 (2019) 240–260.

[47] Z. Wang, Y. Li, C. Ma, X. Yan, D. Jiang, Path-following optimal control of autonomous underwater vehicle based on deep reinforcement learning, Ocean Eng. 268 (2023) 113407.

[48] **zhi Wu, **qiang Bai, F. Hao, G. Cheng, Y. Tang, **uhua Li, Field Complete Coverage Path Planning Based on Improved Genetic Algorithm for Transplanting Robot, Machines 11 (2023) 659.

[49] J. Xue, B. Shen, Dung beetle optimizer: A new meta-heuristic algorithm for global optimization, J. Supercomput. 79 (2023) 7305–7336.

[50] J. Yu, C. Chen, A. Arab, J. Yi, X. Pei, X. Guo, RDT-RRT: Real-time double-tree rapidly-exploring random tree path planning for autonomous vehicles, Expert Syst. Appl. 240 (2024) 122510.

[51] I.A. Zamfirache, R.-E. Precup, R.-C. Roman, E.M. Petriu, Reinforcement Learning-based control using Q-learning and gravitational search algorithm with experimental validation on a nonlinear servo system, Inf. Sci. 583 (2022) 99–120.

[52] Y. Zeng, J. Lyu, R. Zhang, Cellular-Connected UAV: Potential, Challenges, and Promising Technologies, IEEE Wirel. Commun. 26 (2019) 120–127.

[53] X. Zhang, Y. Xu, C. Yu, A.A. Heidari, S. Li, H. Chen, C. Li, Gaussian mutational chaotic fruit fly-built optimization and feature selection, Expert Syst. Appl. 141 (2020) 112976.

[54] D. Zhao, G. Cai, Y. Wang, X. Li, Path Planning of Obstacle-Crossing Robot Based on Golden Sine Grey Wolf Optimizer, Appl. Sci. 14 (2024) 1129.

[55] J. Zheng, M. Ding, L. Sun, H. Liu, Distributed Stochastic Algorithm Based on Enhanced Genetic Algorithm for Path Planning of Multi-UAV Cooperative Area Search, IEEE Trans. Intell. Transp. Syst. 24 (2023) 8290–8303.