2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC) | 979-8-3315-7434-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/COMPSAC65507.2025.00015

Multi-Strategy Enhanced COA for Path Planning in Autonomous Navigation

Yifei Wang[1], Jacky Keung[1], Haohan Xu[2], Yuchen Cao[1], Zhenyu Mao[1,*]
[1]City University of Hong Kong, Hong Kong, China
[2]PetroChina Southwest Oil & Gasfield Company, Chengdu, China
ywang4748-c@my.cityu.edu.hk, Jacky.Keung@cityu.edu.hk,
haohanxu@petrochina.com.cn, cynthcao2-c@my.cityu.edu.hk,
*corresponding author: zhenyumao2-c@my.cityu.edu.hk

***Abstract*—Autonomous navigation is reshaping various domains in people's life by enabling safe and efficient movement in complex environments. Reliable navigation requires path planning algorithms that compute optimal or near-optimal trajectories while satisfying task-specific constraints and ensuring obstacle avoidance. However, existing algorithms struggle with slow convergence and suboptimal solutions, particularly in complex environments, limiting their real-world applicability. To address these limitations, this paper presents the *M*ulti-Strategy Enhanced *C*rayfish *O*ptimization *A*lgorithm (MCOA), a novel approach integrating three strategies: 1) Refractive Learning to enhance diversity and global exploration, 2) Stochastic Centroid-Guided Exploration to balance global and local search, and 3) Adaptive Competition-Based Selection to accelerate convergence and improve solution quality. Experimental results show that MCOA significantly improves the performance of 3D UAV path planning, reducing computation time by 69.2% and trajectory cost by 67.0% compared to 11 baseline algorithms, which demonstrates its effectiveness in autonomous navigation within complex environments.**

***Index Terms*—Autonomous Navigation, Path Planning, Crayfish Optimization Algorithm, Unmanned Aerial Vehicle**

I. Introduction

With rapid advancements in artificial intelligence, sensor technology, and computing power, autonomous navigation has been widely adopted across various domains [1]. It plays a vital role in systems operating in various environments, such as unmanned aerial vehicles (UAVs), mobile robots, and autonomous vehicles, by enabling independent perception, decision-making, and task execution. At its core, path planning is a key process for ensuring safe and efficient movement. It aims to compute an optimal or near-optimal trajectory that meets task-specific constraints while avoiding obstacles, typically through the use of algorithms designed for trajectory optimization [2].

Heuristic and metaheuristic algorithms, such as Genetic Algorithm (GA) [3], Particle Swarm Optimization (PSO) [4], Ant Colony Optimization (ACO) [5], and Crayfish Optimization Algorithm (COA) [6], have proven effective in solving complex path-planning problems through iterative optimization [7]. Building on these algorithms, recent advancements in multi-strategy optimization mechanisms, including multi-point leadership crossover [8] and adaptive competitive swarm optimization [9], have demonstrated promising results in addressing optimization problems.

However, existing algorithms often suffer from premature convergence and suboptimal trajectory, leading to unnecessary detours, high computational costs, and limited adaptability [2]. In addition, the slow convergence of iterative optimization limits their use in real-time applications [10]. These challenges highlight the ongoing trade-off between global exploration and local exploitation, a key obstacle to robust and efficient path planning.

This paper presents MCOA, a novel approach designed to overcome existing limitations by mitigating premature convergence and improving trajectory quality in complex autonomous navigation environments. The key innovations of this work include:

- A refractive based population selection strategy that enhances population diversity, strengthening global exploration and reducing the risk of premature convergence.
- A stochastic centroid-guided exploration strategy that effectively balances local and global search, increasing adaptability and minimizing entrapment in local optima.
- An adaptive competition-based cave selection strategy that adjusts selection pressure over time, facilitating faster convergence during the exploitation phase.

II. Background and Related Works

A. Autonomous navigation and path planning

Autonomous navigation is a key technology that enables autonomous systems to operate independently, safely, and efficiently in unknown environments [11], with wide applications across platforms, particularly UAVs. UAVs have made significant advances in spatial navigation, supporting dynamic target tracking in military reconnaissance [12], precision agriculture with centimeter-level positioning [13], and disaster rescue in complex terrains [14]. Reliable navigation requires path planning algorithms that compute optimal or near-optimal trajectories while satisfying task-specific constraints and ensuring obstacle avoidance [15]. Deterministic algorithms such as A* [16] and Dijkstra algorithm [17] offer reliable results but lack scalability and flexibility in complex scenarios [18], while metaheuristic approaches such as GA, PSO, ACO, and COA

2836-3795/25/$31.00 ©2025 IEEE
DOI 10.1109/COMPSAC65507.2025.00015

have been widely explored as flexible and population-based algorithms for solving complex path planning problems [19].

B. Crayfish optimization algorithm

The Crayfish Optimization Algorithm (COA), proposed by Heming Jia et al. [6], is inspired by the adaptive behaviors of crayfish, a freshwater crustacean that thrives in humus-rich environments. Crayfish exhibit temperature-dependent behavior: they seek shelter in caves at high temperatures and forage actively within the 20–30 °C range. These behaviors are abstracted into the stages of summer resort, competition, and foraging, which correspond to the algorithm's exploration and exploitation phases.

1) Exploration phase (Summer resort stage): When crayfish search for burrows (modeled by $rand < 0.5$), the position of the i-th crayfish in the j-th dimension is updated by Eq.1.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + C_2 \times rand_1 \times (X_{shade} - X_{i,j}^{t}) \tag{1}$$

where

$$C_2 = 2 - (\frac{t}{T}) \tag{2}$$

here, $X_{i,j}^{t}$ and $X_{i,j}^{t+1}$ are the position at iteration t and $t+1$; X_{shade} is a cave location based on current and global best solution, C_2 is an adaptive parameter, t is the current iteration number and T is the max iteration count.

2) Exploitation phase:

- Competition case stage

When $temp > 30$ and $rand \geq 0.5$, crayfish enter the competition for burrows stage, and they will acquire burrows by grabbing, as shown in Eq.3.

$$X_{i,j}^{t+1} = X_{i,j}^{t} - X_{z,j}^{t} + X_{shade} \tag{3}$$

where z is a randomly selected crayfish, calculated by Eq.4, N is population size.

$$z = round(rand_2 \times (N-1)) + 1 \tag{4}$$

- Foraging stage

When $temp \leq 30$, crayfish enter the foraging stage, where they move toward the food source, which is defined as the current global best position X_G. The size of the food is defined as Eq.5.

$$Q = C_3 \times rand_3 \times (\frac{fitness_i}{fitness_{food}}) \tag{5}$$

where C_3 = 3 represents the max food size; $fitness_i$ represents the fitness value of the i-th crayfish; and $fitness_{food}$ represents the fitness value of the food.

When $Q > \frac{C_3+1}{2}$, it means that the food is relatively large, then the crayfish will tear the food, and the mathematical model of tearing is shown in Eq.6.

$$X_{food} = exp(-\frac{1}{Q}) \times X_{food} \tag{6}$$

When the food is torn, the crayfish feeds alternately on the torn food, and the mathematical model of feeding is shown in Eq.7.

$$\begin{aligned} X_{i,j}^{t+1} = X_{i,j}^{t} + X_{\text{food}} \times p \times (&\cos(2\pi \times \text{rand}_4) \\ &- \sin(2\pi \times \text{rand}_5)) \end{aligned} \tag{7}$$

When $Q \leq \frac{C_3+1}{2}$, the crayfish only need to move toward the food and feed directly, and its mathematical model is shown in Eq.8.

$$X_{i,j}^{t+1} = (X_{i.j}^{t} - X_{food}) \times p + p \times rand_6 \times X_{i,j}^{t} \tag{8}$$

Through different feeding mechanisms based on Q, COA approximates the global optimum, where larger food is torn before feeding, while smaller food is consumed directly.

C. Optimizations on COA

To enhance COA's performance on constrained optimization problems, many researchers have conducted optimization studies. Yi Zhang et al. [20] introduced the Halton sequence for uniform population initialization and incorporated the aggregation effect from the marine predator algorithm. Their Enhanced COA (ECOA) achieved a better exploration–exploitation balance and performed well in the Three-Bar Truss and Pressure Vessel Design problems. Heming Jia et al. [21] developed an environment-updating mechanism using water quality factors and ghost antagonism learning, which yielded strong results on the CEC2020 test set. Lakhdar Chaib et al. [22] applied fractional-order chaotic maps and dimensional learning to COA for photovoltaic parameter estimation, demonstrating superior accuracy, stability, and convergence speed.

Despite these advancements, COA still faces inherent challenges. Like many heuristic algorithms, it is prone to premature convergence to local optima, which occurs mainly due to random initialization that may reduce population diversity, and reliance on the global-best solution that can limit exploration and introduce search bias. In addition, the use of fixed competition coefficients often slows the convergence process [23] [24].

III. MCOA

This paper proposes MCOA, an improved COA that integrates three novel strategies to address key limitations of the original COA, notably slow convergence and suboptimal solution quality in complex environments. These strategies focus on enhancing population initialization, exploration capability, and adaptive exploitation, aiming to achieve faster convergence and improved robustness in numerical optimization and path planning tasks.

As shown in Fig.1, MCOA operates in two main phases: the Exploration Phase and the Exploitation Phase. The exploration phase includes the summer resort stage and the Exploitation Phase is further divided into the Competition Stage and the Foraging Stage, resulting in three functional stages overall, where three strategies are applied accordingly. Refraction Learning (Section III-A) improves the initial population by generating reverse solutions (Eq.9) and applying fitness-based selection (Eq.10). Centroid-Guided Exploration (Section III-B) enhances global search during the Exploration Phase (rand < 0.5) via multi-path updates (Eq.11). Adaptive Competition (Section III-C) accelerates convergence in the Competition Stage (temp > 30) using an adaptive range adjustment (Eq.13 and Eq14). In the Foraging Stage, local refinement is performed using a dual-foraging strategy (Eq.7 and Eq.8). The following sections elaborate on the design and function of each component.

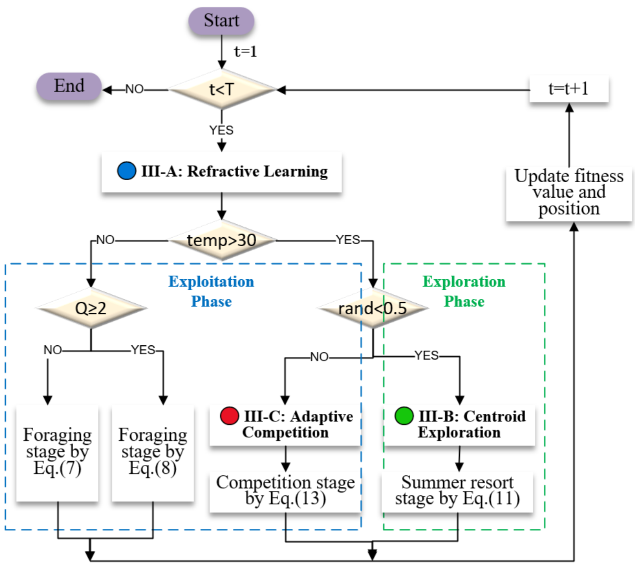

Fig. 1: Flowchart of MCOA

A. Refractive learning for population selection

In the original COA algorithm, the initial population is generated through random sampling, which often results in large variations in individual quality and leads to unstable performance across runs. This can cause poor fitness and slow convergence in the early stages. To address these issues, a refractive learning strategy is introduced to enhance population initialization. As defined in Eq.9, a refractive solution is generated for each individual, and the one with higher fitness is retained. This mechanism improves the overall quality and consistency of the initial population, providing a more robust foundation for the algorithm's subsequent search process.

$$reverse_i = K \cdot (MAX + MIN) - X_i \tag{9}$$

where $reverse_i$ denotes the refractive solution for the individual X_i; K is a row vector of the same dimension as the problem space, containing random values between 0 and 1; MAX and MIN are the maximum and minimum values of the i-th crayfish, respectively. If the refractive solution shows better fitness than the original solution, the original solution is replaced by its refractive solution; otherwise, it remains unchanged, as defined in Eq.10 [25].

$$X_i = \begin{cases} reverse_i, & f(reverse_i) < f(X_i) \\ X_i, & other \end{cases} \tag{10}$$

where $f(reverse_i)$ denotes the fitness value of the opposing solution and $f(X_i)$ denotes the fitness value of the original solution.

B. Stochastic centroid-guided exploration

In the original COA algorithm, position updates during the exploration phase rely solely on global and local best solutions. This limited guidance leads to overly concentrated search directions, which increases the risk of premature convergence and weakens the algorithm's ability to explore promising regions of the solution space [26]. To overcome these limitations, a centroid-guided exploration strategy is proposed. This method introduces stochastic guidance based on multiple centroid references, such as X_{mean}, X_{pmean}, and X_{qmean}, along with random guidance from X_{shade}, to diversify the search behavior. As defined in Eq.11, individuals update their positions using these components, which enhances exploration capability, mitigates local stagnation, and improves the chances of identifying better global solutions.

$$X_{i,j}^{t+1} = X_{i,j}^{t} + C_2 \times rand \times (X_{central(k1,j)} - X_{i,j}^{t}) \tag{11}$$

where $X_{(central(k1,j))}$ contains a total of 6 location information, as shown in Eq.12. $k1$ is a randint(0,5), which is used to decide which location in $X_{central}$ is selected for updating during the location update.

$$X_{central} = \{X_G, X_L, X_{shade}, X_{mean}, X_{pmean}, X_{qmean}\} \tag{12}$$

where X_{mean} is the centroid of all individuals; X_{pmean} and X_{qmean} are the centroids of two randomly selected subsets p and q, respectively. Subset p contains 2 to 5 individuals, while subset q includes 10 to N individuals. X_{shade} denotes a randomly generated position that helps guide the search out of local optima. Fig.2 illustrates six types of random points controlled by the random factor, allowing the algorithm to escape local optima effectively.

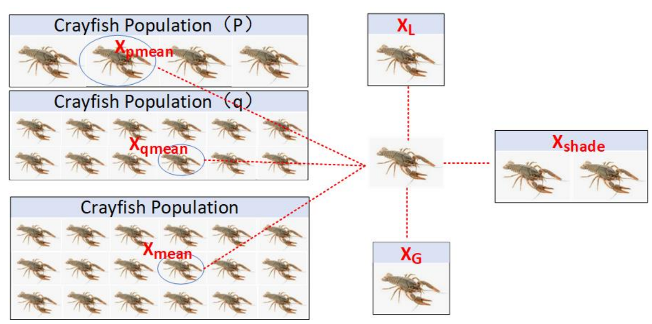

Fig. 2: Six Random Crayfish Populations In Exploration

C. Adaptive competition-based selection

In the original COA algorithm, the exploitation phase updates positions using X_{shade}, $X_{z,j}^{t}$, and randomly selected individuals. However, its fixed competition mechanism cannot dynamically adjust the balance between exploration and exploitation throughout the search process. This often leads to over-reliance on local information, increasing the risk of premature convergence and stagnation in local optima [27]. To address this issue, an adaptive competition strategy is introduced. It incorporates a dynamic competition coefficient CC, which adjusts the search intensity according to the ratio of the current iteration to the maximum number of iterations, as defined in Eq.13. This mechanism encourages exploration in the early phase and gradually shifts toward exploitation in the later phase, enabling faster convergence and improved solution quality.

$$X_{i,j}^{t+1} = X_{shade} + CC \times (X_{i,j}^{t} - X_{z,j}^{t}) \times \epsilon \tag{13}$$

where CC is the adaptive competition coefficient, which is calculated by Eq.14. ϵ is a random variable following the standard normal distribution, which guides individuals to update near the optimal solution and improves solution quality.

$$CC = (1 - \frac{t}{T})^{\frac{2t}{T}} \tag{14}$$

IV. Evaluation

In this section, MCOA algorithm is applied to 3D UAV to verify its practical effectiveness. The performance is evaluated based on two research questions (RQs) as follows:

- RQ1: Is MCOA faster in generating trajectories compared to state-of-the-art algorithms?
- RQ2: Does MCOA generate better trajectories in terms of costs compared to state-of-the-art algorithms?

To systematically evaluate RQ1, the computational time of MCOA is measured and compared with other algorithms. For RQ2, the trajectories generated by MCOA and other baseline algorithms are assessed based on path length cost, threat avoidance cost, altitude constraint cost, flight angle cost, and total cost index. Each algorithm is independently run 30 times to ensure statistical reliability, with a population size of 50 and 1000 iterations per run.

A. Settings

The UAV flight environment is modeled through a composite function (Eq.15-Eq.16) considering realistic environmental obstacles, which subsequently derives four essential constraints(path length, threat avoidance, altitude constraint, and flight angle) formulated in Eq.17 through Eq.28. The experiments involve five UAVs planning trajectories from (150,150,50) to (900,720,150) in the constrained 3D environment.

- UAV flight environment modeling

To simulate realistic flight conditions, the UAV flight environment is modeled using a composite function. The terrain and obstacles are defined as follows:

$$\begin{aligned} z_1(x,y) = 2(\cos x + \sin x) + \sin\left(\sqrt{x^2+y^2}\right) \\ + \cos\left(\sqrt{x^2+y^2}\right) \end{aligned} \tag{15}$$

where $z_1(x, y)$ denotes the underlying terrain of the UAV flight environment. Eq.16 is used to model obstacle factors such as hillsides:

$$z_2(x,y) = \sum_{i=1}^{n} h_i exp(-(\frac{x^2 - x_i}{x_{si}})^2 - (\frac{y^2 - y_i}{y_{si}})^2) \tag{16}$$

where n is the number of slopes, h_i is the height parameter of the slopes, (x_i, y_i) are the coordinates of the center point of the slopes; and x_{si}, y_{si} denote the amount of attenuation of the slopes in the direction of the x-axis and y-axis. n is set to 10, h_i is defined based on the parameters listed in Tab.I.

TABLE I: Peak Parameter Settings

No.	x coordinate	y coordinate	z coordinate	radius
1	400	500	150	30
2	700	150	150	50
3	550	450	150	40
4	350	100	150	50
5	400	650	150	30
6	800	800	150	30
7	750	350	150	70
8	150	350	150	60
9	920	600	150	90
10	920	200	150	50

- Modeling of UAV constraints

In addition to the primary map environment, natural environments typically include buildings, wires, and other obstacles. To simulate these conditions, the UAV flight constraints are modeled as follows.

Path Length Constraint: Path distance is a key factor in UAV path planning, as it directly affects mission efficiency and energy usage [28]. The total path length is computed as shown in Eq.17.

$$F_1(X_i) = \sum_{j=1}^{n-1} L_{p_{i,j}p_{i,j+1}} \tag{17}$$

where $L_{p_{i,j}p_{i,j+1}}$ denotes the distance from node i to node $i+1$, calculated by Eq.18.

$$L_{p_{i,j}p_{i,j+1}} = \text{norm}\left(x_{i+1} - x_i; y_{i+1} - y_i; z_{i+1} - z_i\right) \tag{18}$$

Threat Avoidance Constraint: Avoiding threats such as buildings, trees, and power lines is crucial for UAV safety and mission success. The threat cost is calculated by 19.

$$F_2(X_i) = \sum_{j=1}^{n-1} \sum_{k=1}^{K} T_k(p_{i,j}p_{i,j+1}) \tag{19}$$

where $T_k(p_{i,j}p_{i,j+1})$ denotes flight constraint costs, calculated by Eq.20. where R_k is the radius of the kth cylindrical obstacle, D is the collision region, and d_k is the distance from the obstacle center to the path $L_{p_{i,j}p_{i,j+1}}$. $Penalty$ is the threat cost penalty factor.

$$T_k(p_{i,j}p_{i.j+1}) = \begin{cases} 0, \ d_k \geq D + R_k \\ penalty * (D + R_k - d_k), \\ \quad R_k < d_k < D + R_k \\ \infty, \ d_k \leq R_k \end{cases} \tag{20}$$

Altitude Constraints: In many UAV missions, maintaining flight within a predefined altitude range is necessary to comply with airspace regulations or mission parameters. This constraint is modeled by Eq.21.

$$F_3(X_i) = \sum_{j=1}^{n} H_{i,j} \tag{21}$$

where $H_{i,j}$ denotes the cost of the height of the X_i location, calculated by Eq.22. where $penalty$ is the penalty coefficient, $h_{i,j}$ is the UAV's current altitude, h_{min} is the minimum allowed altitude, and h_{max} is the maximum allowed altitude.

$$H_{i,j} = \begin{cases} penalty * (h_{i,j} - h_{max}), \ h_{i,j} \geq h_{max} \\ 0, \ h_{min} < h_{i,j} < h_{max} \\ penalty * (h_{min} - h_{i,j}), \\ \quad 0 < h_{i,j} \leq h_{min} \infty, \ h_{i,j} \leq 0 \end{cases} \tag{22}$$

Flight Angle Constraint: The UAV's horizontal turn angles $\alpha_{i,j}$ and vertical pitch angles $\beta_{i,j}$ are constrained to ensure smooth trajectories. The flight angle cost is calculated by Eq.23. where a_1 and a_2 denote the penalty coefficients for the UAV's horizontal turn angle and vertical pitch angle constraints respectively. where $\alpha_{i,j}$ and $\beta_{i,j}$ are calculated by Eq.24 and Eq.25.

$$F_4(X_i) = a_1 \sum_{j=1}^{n-2} \alpha_{i,j} + a_2 \sum_{j=1}^{n-1} |\beta_{i,j} - \beta_{i,j-1}| \tag{23}$$

$$\alpha_{i,j} = arctan \frac{L_{p'_{i,j} p'_{i,j+1}} \times L_{p'_{i,j} p'_{i,j+2}}}{L_{p'_{i,j} p'_{i,j+1}} \cdot L_{p'_{i,j} p'_{i,j+2}}} \tag{24}$$

$$\beta_{i,j} = arctan \frac{Z_{i,j+1} - Z_{i,j}}{||L_{p_{i,j} p_{i,j+1}}||} \tag{25}$$

where $L_{p'_{i,j} p'_{i.j+1}}$ is their projections on the plane, calculated by Eq.26. where k is the unit vector in the positive direction of the axis.

$$L_{p'_{i,j}, p'_{i,j+1}} = k \times \left(L_{p_{i,j}, p_{i,j+1}} \times k\right) \tag{26}$$

Total Cost Index: The total cost index is computed as a weighted sum of the four normalized constraint costs, as defined in Eq.27.

$$F(X_i) = \sum_{k=1}^{4} F_{\text{norm}}(X_i) \tag{27}$$

where $F_{\text{norm}}(X_i)$ is normalized using a log transformation based min-max normalization, as defined in Eq.28.

$$F_{\text{norm}}(X) = \frac{\log(F(X) + 10^{-5}) - \min(\log(F(X) + 10^{-5}))}{\max(\log(F(X) + 10^{-5})) - \min(\log(F(X) + 10^{-5}))} \tag{28}$$

B. Results

Tab.II presents the average cost metrics for each algorithm in the UAV path planning task, including path length, threat avoidance, altitude constraint, flight angle, and total cost index. MCOA achieves the lowest total cost of 0.75, which is 39.5% lower than the second-best result (HLOA: 1.24) and 67.0% below the overall average. In detail, MCOA achieves a path length of 5206.1, which is 8.4% shorter than the average (5682.6). The threat avoidance cost is 31.4, 11.8% lower than the average (35.6). For altitude constraint, it records 528.7, compared to the average of 813.6. It also achieves a flight angle cost of zero, indicating smoother trajectory planning than other algorithms. Fig.3 illustrates the path trajectories planned by the MCOA algorithm. In terms of time consumption, also reported in Tab.II, MCOA required only 13.8 seconds, the fastest among all tested algorithms. This represents a 69.2% improvement in computational efficiency compared to the average of 44.8 seconds.

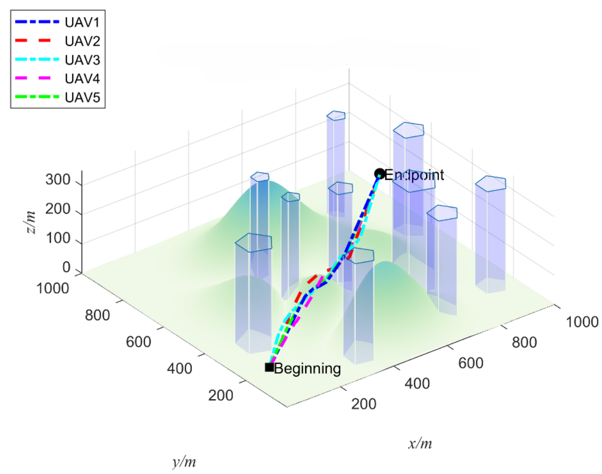

Fig. 3: UAV Trajectory of MCOA

C. Discussions

In the 3D UAV path planning experiments, MCOA demonstrated a clear advantage in generating high-quality trajectories under complex environmental conditions. It reduced the total cost by 67.0% compared to the average, reflecting improvements in path efficiency and mission duration, which are critical for time-sensitive UAV applications. Moreover, MCOA produced trajectories with minimal directional variation, enhancing overall flight stability. These characteristics make it well-suited for dynamic missions such as low-altitude reconnaissance and precision aerial delivery.

While MCOA performs well overall, the experimental results also reveal areas for improvement in complex environments. Compared to COA, MCOA shows limited improvement in managing threat-related and altitude constraint costs, particularly in high-obstacle-density regions. In such scenarios, its altitude response is slightly slower. For example, when

TABLE II: Performance Comparison of UAV Path Planning Algorithms

Metric / Algorithm	DBO	WOA	NRBO	LEA	HLOA	HEOA	GOOSE	CPO	GWO	BKA	COA	**MCOA**
Path Length	5806.1	5478.6	5525.8	5693.8	5447.5	5356.9	6005.5	6065.8	5910.2	6161.3	5533.2	**5206.1**
Threat Avoidance	42.1	40.4	31.6	18.8	39.9	**14.6**	38.3	22.4	26.8	89.8	30.5	31.4
Altitude Constraint	506.5	993.7	1306.6	1656.9	617.6	685.7	902.1	977.5	**305.8**	765.5	516.7	528.7
Flight Angle	488.5	490.9	676.5	1145.2	0.0	81.3	799.8	291.2	491.2	437.0	473.9	**0.0**
Total Cost Index	2.48	2.52	2.61	2.67	1.24	1.51	2.99	2.76	2.22	3.49	2.03	**0.75**
Time (s)	14.3	13.8	15.5	16.3	17.6	181.1	44.7	42.2	84.5	78.2	15.9	**13.8**

encountering regions with abrupt terrain elevation, the UAVs guided by MCOA exhibit an average delay of 0.5 seconds in altitude adjustment compared to those guided by COA, which could affect obstacle avoidance performance in rapid-response flight missions. Enhancing the cost modeling and adaptation strategies related to threats and altitude constraints could further improve MCOA's robustness in complex and dynamic planning tasks.

V. Conclusion and Future Works

This paper introduces MCOA, a multi-strategy enhanced COA algorithm tailored for autonomous navigation, integrating refractive learning, stochastic centroid-guided exploration, and adaptive competition-based selection. By refining population diversity, balancing global and local search, and accelerating convergence, MCOA produces more optimal paths with faster computation. Comparative evaluations against 11 baseline algorithms demonstrate that, while adhering to real-life constraints, MCOA reduces total path cost by 67.0% and shortens computation time by 69.2% in 3D UAV scenarios. These results highlight MCOA as a promising solution for real-world autonomous navigation applications.

References

[1] S. Nahavandi, R. Alizadehsani, D. Nahavandi, S. Mohamed, N. Mohajer, M. Rokonuzzaman, and I. Hossain, "A comprehensive review on autonomous navigation," *arXiv preprint arXiv:2212.12808*, 2022.

[2] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Systems with Applications*, vol. 227, p. 120254, 2023.

[3] S. Mirjalili and S. Mirjalili, "Genetic algorithm," *Evolutionary algorithms and neural networks: Theory and applications*, pp. 43–55, 2019.

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

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

[6] H. Jia, H. Rao, C. Wen, and S. Mirjalili, "Crayfish optimization algorithm," *Artificial Intelligence Review*, vol. 56, no. Suppl 2, pp. 1919–1979, 2023.

[7] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Computational intelligence for multimedia big data on the cloud with engineering applications*, pp. 185–231, 2018.

[8] X. Zhang, S. Wang, K. Zhao, and Y. Wang, "A salp swarm algorithm based on harris eagle foraging strategy," *Mathematics and Computers in Simulation*, vol. 203, pp. 858–877, 2023.

[9] W. Huang and W. Zhang, "Multi-objective optimization based on an adaptive competitive swarm optimizer," *Information Sciences*, vol. 583, pp. 266–287, 2022.

[10] B. Liao, F. Wan, Y. Hua, R. Ma, S. Zhu, and X. Qing, "F-rrt*: An improved path planning algorithm with improved initial solution and convergence rate," *Expert Systems with Applications*, vol. 184, p. 115457, 2021.

[11] R. C. Shit, "Precise localization for achieving next-generation autonomous navigation: State-of-the-art, taxonomy and future prospects," *Computer Communications*, vol. 160, pp. 351–374, 2020.

[12] M. Gargalakos, "The role of unmanned aerial vehicles in military communications: application scenarios, current trends, and beyond," *The Journal of Defense Modeling and Simulation*, vol. 21, no. 3, pp. 313–321, 2024.

[13] P. Velusamy, S. Rajendran, R. K. Mahendran, S. Naseer, M. Shafiq, and J.-G. Choi, "Unmanned aerial vehicles (uav) in precision agriculture: Applications and challenges," *Energies*, vol. 15, no. 1, p. 217, 2021.

[14] T. Li and H. Hu, "Development of the use of unmanned aerial vehicles (uavs) in emergency rescue in china," *Risk Management and Healthcare Policy*, pp. 4293–4299, 2021.

[15] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mechanical systems and signal processing*, vol. 100, pp. 482–500, 2018.

[16] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[17] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[18] L. Wenzheng, L. Junjun, and Y. Shunli, "An improved dijkstra's algorithm for shortest path planning on 2d grid maps," in *2019 IEEE 9th international conference on electronics information and emergency communication (ICEIEC)*. IEEE, 2019, pp. 438–441.

[19] H. S. Yahia and A. S. Mohammed, "Path planning optimization in unmanned aerial vehicles using meta-heuristic algorithms: A systematic review," *Environmental Monitoring and Assessment*, vol. 195, no. 1, p. 30, 2023.

[20] Y. Zhang, P. Liu, and Y. Li, "Implementation of an enhanced crayfish optimization algorithm," *Biomimetics*, vol. 9, no. 6, p. 341, 2024.

[21] H. Jia, X. Zhou, J. Zhang, L. Abualigah, A. R. Yildiz, and A. G. Hussien, "Modified crayfish optimization algorithm for solving multiple engineering application problems," *Artificial Intelligence Review*, vol. 57, no. 5, p. 127, 2024.

[22] L. Chaib, M. Tadj, A. Choucha, F. Z. Khemili, and E.-F. Attia, "Improved crayfish optimization algorithm for parameters estimation of photovoltaic models," *Energy Conversion and Management*, vol. 313, p. 118627, 2024.

[23] S. Jafari, O. Bozorg-Haddad, and X. Chu, "Cuckoo optimization algorithm (coa)," *Advanced optimization by nature-inspired algorithms*, pp. 39–49, 2018.

[24] G. Li, T. Zhang, C.-Y. Tsai, Y. Lu, J. Yang, and L. Yao, "Modified crayfish optimization algorithm with adaptive spiral elite greedy opposition-based learning and search-hide strategy for global optimization," *Journal of Computational Design and Engineering*, vol. 11, no. 4, pp. 249–305, 2024.

[25] B. Tu, F. Wang, and X. Han, "3d path planning for uav based on a hybrid algorithm of marine predators algorithm with quasi-oppositional learning and differential evolution," *Egyptian Informatics Journal*, vol. 28, p. 100556, 2024.

[26] J. Pierezan and L. D. S. Coelho, "Coyote optimization algorithm: a new metaheuristic for global optimization problems," in *2018 IEEE congress on evolutionary computation (CEC)*. IEEE, 2018, pp. 1–8.

[27] P. Binev, W. Dahmen, and R. DeVore, "Adaptive finite element methods with convergence rates," *Numerische Mathematik*, vol. 97, pp. 219–268, 2004.

[28] S. Ghambari, M. Golabi, L. Jourdan, J. Lepagnot, and L. Idoumghar, "Uav path planning techniques: a survey," *RAIRO-Operations Research*, vol. 58, no. 4, pp. 2951–2989, 2024.