Knowledge-Based Systems 317 (2025) 113481

Contents lists available at ScienceDirect

Knowledge-Based Systems

Journal homepage: www.elsevier.com/locate/knosys

Enhancing unmanned marine vehicle path planning: A fractal-enhanced chaotic grey wolf and differential evolution approach

Chaoyang Zhu [a], Yassine Bouteraa [b,c,*], Mohammad Khishe [e,d,g], Diego Martín [f], Francisco Hernando-Gallego [f], Thavavel Vaiyapuri [h]

[a] Institute of Social Innovation and Public Culture, Communication University of China, Beijing, 100024, PR China
[b] Department of Computer Engineering, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj 11942, Saudi Arabia
[c] Applied Science Research Center, Applied Science Private University, Amman, Jordan
[d] Department of Biomedical, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, 602 105, India
[e] Department of Computer Science, Escuela de Ingeniería Informática de Segovia, Universidad de Valladolid, Segovia, 40005, Spain
[f] Advanced Technologies for Medicine and Signals (ATMS), ENIS, University of Sfax, Tunisia
[g] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India

ARTICLE INFO

Keywords:
Path planning
Unmanned marine vehicle
Fractals chaotic maps
Grey wolf optimizer
Differential evolution

ABSTRACT

Efficient path planning is challenging for optimizing the trajectory of uncrewed marine vehicles navigating complex environments. However, when the global optimum is zero, path planning optimization encounters a significant challenge, a major shortcoming of the grey wolf optimizer (GWO). This study intentionally integrates multiple approaches to present a comprehensive methodology called fractal-enhanced chaotic GWO (FSCGWO) in conjunction with differential evolution (DE) to fill this research gap. This method uses DE to strengthen the local search or exploitation phases, chaotic maps to improve the exploration phase, and fractals to fine-tune the transition between the two phases. In addition to testing against 46 sophisticated benchmark maps, this study carries out practical experimentation over commonly utilized meta-heuristic algorithms to comprehensively evaluate the proposed hybrid model's performance (FSCGWO-DE). This thorough evaluation demonstrates notable advancements in unmanned marine vehicle path planning. The evaluation criteria include path length, consistency, time complexity, and success rate. These metrics illustrate the statistical significance of the novel methodology's improvements. The study demonstrates that FSCGWO can precisely identify the best routes in given test maps, offering insightful information for developing path planning optimization—especially concerning unmanned marine vehicles.

1. Introduction

Unmanned Marine Vehicles (UMVs) have become essential tools for various applications relating to underwater research [1], autonomous vehicles [2], environmental monitoring [3], navigation in urban areas [4], and air-ground integrated infrastructure design. Their efficiency and safety depend on path planning (PAPL) and safety [5], algorithms, determining a course that completes all tasks while avoiding hazards and minimizing risks [6–8]. This can be very difficult to accomplish in marine environments because moving parts such as currents, random obstacles, and shifting environmental conditions add to the difficulty of navigation [9–11]. With the more prevalent use of UMVs in real-world situations, there is an increasing need for reliable and versatile path-planning solutions, which is currently being met by advanced optimization techniques [12,13].

Even with advanced technological developments, the existing path planning methods cannot adequately balance exploring and exploiting new and existing search spaces [14–16]. This deficiency can result in suboptimal paths being created, or the system can converge to local optima instead. Traditional methods such as probabilistic roadmap methods (PRM) [17] and metaheuristic strategies like the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are effective in dealing with static environments; however, they often struggle in dynamic and high-dimensional marine settings [18]. In addition, many of

* Corresponding author.
E-mail addresses: zcy0919pqy@cuc.edu.cn (C. Zhu), yassine.bouteraa@tdtu.edu.vn (Y. Bouteraa), m.khishe@alumni.iust.ac.ir (M. Khishe), diego.martin.andres@uva.es (D. Martín), fhernando@eva.es (F. Hernando-Gallego), thavavel.vaiyapuri@psau.edu.sa (T. Vaiyapuri).

https://doi.org/10.1016/j.knosys.2025.113481
Received 20 May 2024; Received in revised form 21 March 2025; Accepted 1 April 2025
Available online 2 April 2025

these methods require significant amounts of time and resources in compute, greatly diminishing their effectiveness in real-time scenarios where speed is critical [19,20]. These shortcomings, while challenging, present an opportunity to create new methods to solve complex problems in marine ecosystems reliably and efficiently.

To address these issues, this study attempts to enhance UMV path planning by formulating a new optimization approach using nature-inspired algorithms to achieve shorter, smoother, and collision-free trajectories. Our approach addresses conventional methods with bio-inspired techniques by increasing search-space diversity, improving solution-sculpting, and adaptability to changeable environments. We carried out many controlled experiments and statistical analyses, proving that the strategy is much better than the current solutions. This serves as an answer to the problem of real-time navigation of UMVs.

The key contributions of this work are as follows:

1. Increase exploration using chaotic maps, preventing premature convergence and increasing search diversity.
2. Improve exploitation through DE mutation and crossover, improving solutions to optimal path planning.
3. Employ a fractal-based multi-scale search strategy that adapts to the dynamic marine environment.
4. Enhance computational efficiency for real-time performance and adaptability in complicated situations.

When FEC is integrated into GWO and DE, the approach improves existing methods. It solves the researched gaps to achieve optimal path lengths, smoother trajectories, high success rates, low computational time, and effectiveness for UMV navigation in complex and dynamic environments.

The rest of the paper is structured as follows: Section 2 presents an overview of the literature, Section 3 represents the preliminary, Section 4 describes the proposed methodology, Section 5 contains results and their comparison, Section 6 contains a discussion on the results and their implications, and Section 7 provides a summary with recommendations for further research.

2. Related work

The development of path planning for UMVs has received particular focus owing to the importance of efficient and safe navigation in complex marine environments. Kavraki et al. [17] conducted one of the early attempts, which proposed the PRM that samples the configuration space and attempts to connect collision-free points to form a graph of feasible paths. In high-dimensional static settings, PRM is effective, but it is not useful in the presence of changing dynamic obstacles, yielding suboptimal, coarse paths. This poses severe limitations for real-time UMV navigation. To overcome these difficulties, other researchers have resorted to nature-inspired metaheuristic algorithms that have proven to be flexible in the face of complex optimization problems [21].

Among the techniques considered, swarm-based methods appear to be the most developed. Wang et al. [22] utilized ant colony optimization (ACO) for robot pathfinding, capitalizing on its global search capabilities. Similarly, Rodan et al. [23] modified PSO for unmanned surface vehicles and obtained good results in less complex cases. However, both ACO and PSO have problems with scalability in large or dynamic environments because ACO tends to be computationally intensive and PSO is susceptible to local minima. In a different approach, Hamad et al. [24] introduced the GWO based on wolf pack hunting strategies to UAV path planning to manage the balance between exploration and exploitation. GWO provides a strong framework without a global optimum—an increasingly prevalent situation in marine contexts due to limited navigable routes—its effectiveness weakens, demonstrating a need for refinement.

Further enhancement may also be possible using evolutionary algorithms. Li et al. [25] implemented GA to develop feasible trajectories for UMV due to their global optimization searching features. Unfortunately, GA's convergence issues and the need for precise parameter-value tuning make it ineffective in real-time situations. Rada et al. [26] investigated DE for UAV path planning and showed remarkable performance in local search with the mutation and crossover operators. Despite DE's exceptional performance in exploitation, its efficiency in exploration is significantly impacted by its chosen parameters, often requiring large powered populations to yield dependable outcomes. In addition to these efforts, Kusuru et al. [27] implemented the harmony search algorithm (HSA) in mobile robot navigation and noted the lack of sufficient diversity maintaining strategies, but this strategy does not converge in large search areas.

Other firefly algorithms (FA) [28] and bacterial foraging optimization (BFO) by Wang et al. [29] also focus on multi-robot systems. FA is straightforward to use but lacks scalability, while BFO is suitable for dynamic environments but has slow convergence. Ni et al. [30] used parallelism to improve robot path planning with the artificial bee colony (ABC) algorithm, but its exploitation capability is minimal. Yu et al. [31] used Cuckoo's algorithm with reinforcement learning to enhance exploration for UAVs, but it is prone to premature converging without precise adjustments.

These studies demonstrate that there is still a gap in the literature when it comes to providing a balanced solution for all aspects of UMV path planning [32,33]. Table 1 captures the key methods and highlights the remaining gaps: lack of balance between exploration and exploitation, strong susceptibility to local optima, and high computational cost in dynamic, high-dimensional environments. The "no free lunch" theorem [34] stresses that no single metaheuristic performs best in every case, motivating the need to combine approaches with differing strengths. Our approach improves these gaps by fusing the local search capabilities of GWO and DE's evolutionary update with chaotic and fractal strategies.

The examined methods show several research gaps in current path planning algorithms, particularly concerning the tradeoff between exploration and exploitation, dealing with dynamic, changing environments, convergence speed, and computational efficiency. Single heuristic approaches such as GAs, ACO, and GWO have problems with either local search, a tendency towards premature convergence, or very high computational costs. The performance of hybrid approaches is better, but these approaches also have limitations concerning adaptability and optimization of search as follows:

- **Imbalance in exploration and exploitation of search space**

Most existing metaheuristic algorithms have a tough time splitting their focus between prospecting, which is searching new areas, and exploitation, which is refining the previously attained solutions, leading to inefficient path planning.

- **Local optima convergence**

Many researchers use conventional algorithms that often converge too fast. This means that they get stuck in local optima instead of traversing towards what would be interpreted as the global best path.

- **Dynamic and complex marine environment problems**

Marine environments are of the real-world variety, meaning they are highly dynamic and have unpredictable obstacles, currents, and other environmental changes, which makes static path planning problem techniques ineffective.

- **Real-time efficient computational problem**

Many optimization techniques must process a lot of data simultaneously, making them not very applicable to real-time UMV navigation

Table 1
Disadvantages, advantages, and research gaps of reviewed PAPLs.

Method	Disadvantages	Advantages	Research Gaps
PRM [17]	Struggles with dynamic obstacles; Paths may not be optimal in terms of smoothness and distance	Efficient in high-dimensional spaces; Works well for static environments	Lacks adaptability to dynamic environments; Struggles with real-time obstacle avoidance
ACO [18]	Computationally expensive; May get stuck in local optima	Suitable for multi-agent systems; Performs well in dynamic environments	Poor scalability in large environments; Sensitive to parameter settings
GA [19]	Slow convergence; Requires extensive parameter tuning	Good global search capability; Can handle complex environments	Needs better balance between exploration and exploitation; High computational cost
PA [20]	Scalability issues; Sensitive to parameter tuning	Easy to implement; Performs well in continuous optimization	Requires enhancements for high-dimensional and dynamic path planning
HBA [21]	Prone to premature convergence; Struggles with local optima	Fast convergence; Simple implementation	Needs improved local search mechanisms to prevent stagnation
CBA [22]	May converge prematurely; Less effective for fine-tuning solutions	Strong exploration ability; Avoids local optima in some cases	Needs improved exploitation to refine path solutions
BFO [23]	Slow convergence; High computational cost in large-scale problems	Adaptable to dynamic environments; Good exploration capability	Lacks effective fine tuning for path optimization
DE [24]	Poor exploration ability; May require large populations for effectiveness	Strong local search capability; Robust against parameter tuning	Needs integration with other algorithms to improve exploration
GWO [25]	Suffers from stagnation; Sensitive to parameter settings	Balances exploration and exploitation; Simple to implement	Struggles in complex environments with dynamic constraints; Requires modifications to enhance search capabilities
PSO [23]	May still struggle with exploration-exploitation balance; Computational overhead	Improved search efficiency; Combines strengths of multiple methods	Further improvements are needed to optimize trade-offs between efficiency and accuracy

tasks where decisions must be made quickly.

- Smoothing the path and collision avoidance

Some existing methods are unsatisfactory in their functioning. They produce far too complex or straightforward paths, causing the UMV to move inefficiently and unsafely.

- Moving boundaries problems of high dimensions

As the environment becomes more complex, performance decreases when using traditional algorithms, meaning that more robust methods are needed to deal with these high-dimensional spaces.

3. Preliminary

This section introduces the fundamental concepts and methodologies that form the basis of the proposed approach. It provides an overview of key optimization techniques, including DE and GWO, and their relevance to UMV path planning.

3.1. Differential evolution (DE)

DE is a resilient optimization algorithm that falls into the category of evolutionary algorithms [26]. DE demonstrates exceptional proficiency in resolving intricate optimization problems. The algorithm progressively improves a group of potential solutions by utilizing the disparity between randomly chosen individuals, referred to as "mutation." This disparity is merged with another entity using crossover and selection methods.

3.1.1. Mathematical representation

Define X_t as the population at generation t, $X_t = (x_{1,t}, x_{2,t}, \ldots, x_{NP,t})$, and NP represents the size of the population. The mutation operation for each target individual $x_{i,t}$ is defined as Eq. (1) [26]:

$$v_{i,t} = x_{r_1,t} + F\,(x_{r_2,t} - x_{r_3,t}) \tag{1}$$

In this context, r_1, r_2, and r_3 represent indices selected randomly and distinct from i, and F is a scaling factor determining the differential variation's degree of amplification.

The following crossover process merges the trial vector $v_{i,t}$ with the target vector $x_{i,t}$ to produce a novel solution $u_{i,t}$. The selection process assesses whether $u_{i,t}$ supplants $x_{i,t}$ in the subsequent generation are contingent upon the objective function's minimization.

3.2. Grey wolf optimization (GWO)

GWO is a nature-inspired optimization algorithm that draws inspiration from the social hierarchy and hunting behavior of grey wolves. Proposed by Mirjalili et al. [27], the GWO imitates the hierarchical organization of a wolf pack and utilizes surrounding prey and hunting tactics to improve the group of potential solutions progressively.

3.2.1. Mathematical representation

Denote X_t as the population at iteration t, where $X_t = (x_{1,t}, x_{2,t}, \ldots, x_{N,t})$ and N represents the pack size. The position update for each wolf $x_{i,t}$ is determined by Eq. (2) [27]:

$$x_{i,t+1} = \frac{x_{\alpha,j} + x_{\beta,j} + x_{\delta,j}}{3} + \mathrm{rand}\,A \tag{2}$$

In this context, the symbol j denotes the index of the components in the solution vector, while A is a variable parameter that governs the balance between exploration and exploitation. The term "rand" introduces randomness to the search space, promoting diversity.

3.3. Probabilistic roadmap method (PRM)

PRM is a prevalent technique utilized in motion planning. The PRM algorithm generates a roadmap, a graph depicting the connectivity of the configuration space. This performance is achieved by randomly sampling points and linking them using collision detection. It offers a streamlined method for devising viable routes for robots and other self-governing systems in intricate and multidimensional surroundings.

4. Methodology

This section provides an overview of the FECGWO, shows how the PAPL problem is structured, illustrates the process of generating feasible paths using FECGWO, and discusses optimizing the global PAPL with DE.

4.1. The task of defining PAPL

A strategy, denoted as ST, is devised for the UMV by representing it as a two-dimensional map consisting of a set of fixed obstacles, O_k, in

which j is an integer that ranges from 1 to T. Additionally, there is a UMV V in the vicinity, and its precise position is denoted by its location L_i (x, y). The UMV is hypothesized to have a circular shape, resembling a disk with a radius referred to as "RD." The objective is to identify a sequence of motion operations, referred to as Ω, enabling the UMV to travel from Ls (x, y) to the intended endpoint LE (x, y), illustrated in Fig. 1.

Soft computing offers The benefit of hybridization, which incorporates various methodologies. Several experiments were carried out using different techniques to tackle the PAPL problem. We investigated evolutionary methodologies separately, explicitly focusing on FECGWO and DE. Nevertheless, finding the best routes was difficult when we just used the FECGWO method. Although viable paths were generated, they did not match the set optimization criteria. In addition, we performed studies where we only used DE as the producer of path plans. However, there were issues with how well this strategy prevented collisions along the UMV's route. Nevertheless, it showed signs of improvement when it began down a promising road.

We propose a nature-inspired technique that integrates rotation primitives with forward movement primitives to address these challenges. There are two primary steps to the approach used in this study. Connecting unencumbered configurations from the origin to the desired destination position creates an ideal path free of obstacles. When used in local search approaches, the FECGWO method makes linking previously unconnected path segments easier. On top of that, we enhance the initial path's duration and smoothness using a global optimization technique known as DE. Paths that are short, consistent, and collision-free are given priority in the fitness function that is described in this study.

4.2. Proposed optimizer: FECGWO

The FECGWO method incorporates chaotic maps into the location update calculations of the GWO to augment its exploration capabilities. In addition, a search approach that relies on fractals is used to improve the search space exploration.

4.2.1. Rationale for chaotic maps

Incorporating chaos maps inside the GWO is motivated by enhancing the system's ability to explore and exploit [20]. The application of chaotic maps in the investigation of the search space of the GWO algorithm shows potential because of their responsiveness to initial settings and pseudo-random behaviour [36]. This potential use enables a variety of search procedures, preventing early convergence and enhancing the algorithm's capacity to deviate from local optima.

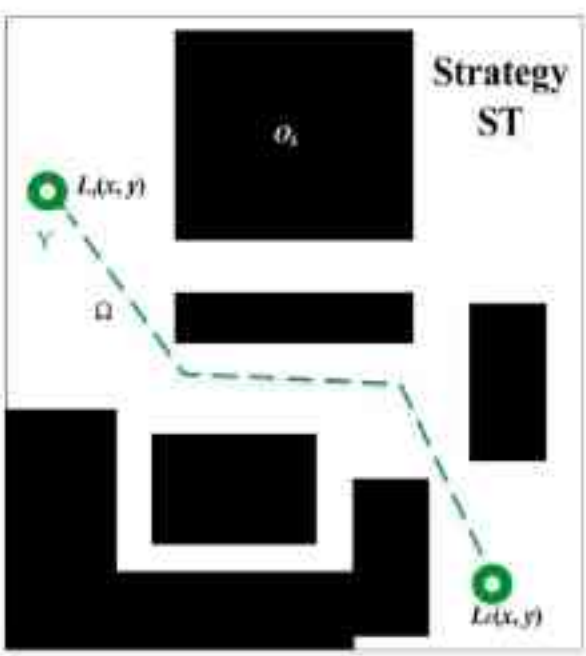


Fig. 1. The approach our research takes in defining the issue of PAPL. The UMV V is represented as a circle with a radius equal to RD.

4.2.2. Execution of chaotic maps

A chaotic map within the GWO approach will be incorporated by altering the equations to modify the positions. In the standard GWO, the positions of the searching agents, known as wolves, are updated by considering the locations of the alpha, beta, and delta wolves. These wolves embody the most optimal solutions found up to this point. Adding a chaotic variable to the formulas employed for updating the locations of search agents is expected to alter their trajectory. This improvement aims to enhance exploring and exploiting the search space. The location update of each wolf in the GWO is represented as follows [37]:

$$L^{(t+1)} = L^{(t)} + \theta \odot \varphi \tag{3}$$

In the next iteration, the location of a wolf is indicated by $L^{(t+1)}$, while its present location is represented by $L^{(t)}$. The symbol ⊙ denotes the operation of multiplying corresponding elements. Eqs. (4) and (5) are used to calculate the matrices θ and φ [37].

$$\theta = 2a \times rand_1 - a \tag{4}$$

$$\varphi = 2 \times rand_2 \tag{5}$$

Where 2a denotes a parameter that gradually declines linearly from 2 to 0 as the iterations progress. Additionally, $rand_1$ and $rand_2$ represent random vectors within the range of [0, 1]. The location update equation of the FECGWO incorporates chaotic maps in Eq. (6):

$$L^{(t+1)} = L^{(t)} + CM^{(t)} \odot (\theta \odot \varphi) \tag{6}$$

The variable $CM^{(t)}$ denotes the value of chaotic maps at iteration t. Various chaotic maps $CM^{(t)}$, as depicted in Table 3 and Fig. 2, can be produced [38]. The mathematical expression for this strategy is as follows:

$$CM^{(t+1)} = \mu \times CM^{(t)} \times (1 - CM^{(t)}) \tag{7}$$

All of the graphical depictions in this study have been set to a reference point of 0.7. Table 3 displays the acronym for each FECGWO, which needs to be recognized.

4.2.3. The rationale behind fractals

Complex fractals are built up by repeatedly recreating simple patterns over various sizes. The GWO implementation aims to incorporate a multi-scale search strategy that takes advantage of fractals' self-similarity properties to explore the search space more effectively.

4.2.4. Application of fractals

One way to include fractals into the GWO is to use a multi-scale search strategy, where search agents, called wolves, investigate the search domain at different scales, similar to how a fractal's navigation across different scales works. This feature could let search agents explore

Table 3
The implemented chaotic maps. Every map has a range of 0 to 1.

Name	Chaotic map
Singer	$L_{t+1} = 1.07 \times (7.86 \times L_t - 23.31 \times L_t^2 + 28.75 \times L_t^3 - 13.3 \times L_t^4)$
Quadratic	$L_{t+1} = L_t^2 - 1$
Gauss/ mouse	$L_{t+1} = \begin{cases} 1 & L_t = 0 \\ 1/\mathrm{mod}(L_t, 1) & \text{otherwise} \end{cases}$
Logistic	$L_{t+1} = 4L_t \times (1 - L_t)$
Tent	$L_{t+1} = \begin{cases} 1.45L_t & L_t < 0.7 \\ 3.33(1 - L_t) & L_t \geq 0.7 \end{cases}$
Bernoulli	$L_{t+1} = 3L_t(\mathrm{mod}1)$

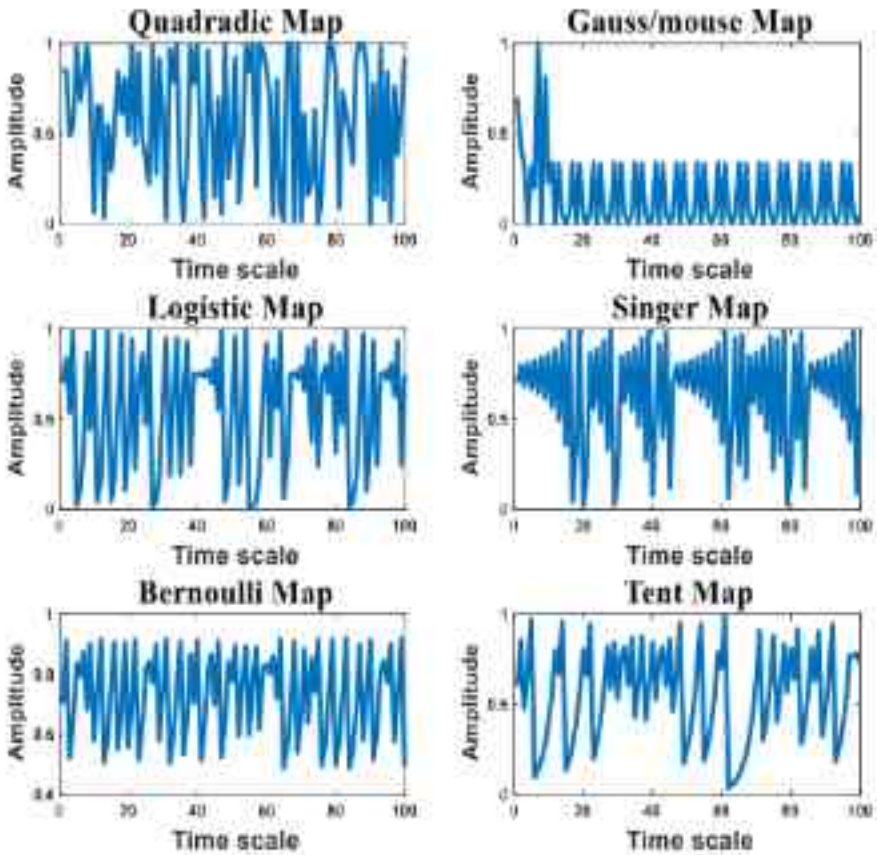


Fig. 2. The implemented chaotic maps.

Table 2
FECGWO labels.

Maps	Gauss/mouse	Quadratic	Singer	Logistic	Tent	Bernoulli
labels	FECGWO1	FECGWO2	FECGWO3	FECGWO4	FECGWO5	FECGWO6

and exploit the search area at several degrees of detail, revealing optimal solutions that are not always obvious when looking at the space at one scale.

An enhanced search method based on multi-scale fractals is created to improve the GWO's exploitation capabilities. The self-similarity of a particular fractal pattern guides the multi-scale search space investigation.

$$L^{(t+1,M)} = L^{(t,M)} + N^{(M)} \odot \left(L^{M}_{best} - L^{(t,M)}\right) \tag{6}$$

Where M is the level of the scale in this iteration, we take into account the wolf's position (represented by $L^{(t,M)}$), as well as its position in

Algorithm 1
Pseudocode of FDCGWO

```
Input: Objective function f(x), x = [x_1, x_2, ..., x_d]
  Number of wolves N,
  Max number of iterations T,
  Scales δ
  Output: Optimal solution P*
  1: Initialize the wolf population and calculate the fitness of each wolf
  2: Identify the alpha, beta, and delta wolves
  3: Initialize the chaotic variable CV^0 using a random value
  4: for t = 1 to T do
  5:     for each wolf do
  6:         Update the chaotic variable CV^t using the chaotic map
  7:         Update the position of the wolf using chaotic maps and the positions of alpha, beta, and delta wolves
  8:         Update the wolf's position and fitness if the new position is better
  9:     end for
  10:    for j = 1 to δ do
  11:        Employ a fractal-based multi-scale search for each wolf using scale δ
  12:    end for
  13:    Update alpha, beta, and delta wolves if better solutions are found
  14: end for
  15: Return the best solution P*
```

the following one (represented by $L^{(p+1,M)}$). The scale factor is denoted by $N^{(M)}$ while the most well-known position at that particular scale level is represented by L^{M}_{best}. Thanks to this capability, the algorithm may search at several resolutions, which could help it find optimal solutions that are not obvious at a single scale. The pseudocode of FECGWO is shown in Algorithm 1 and the framework of the proposed algorithm is shown in Fig. 3.

4.3. Generating paths using FECGWO

This section will review the FECGWO and its potential use for PAPL optimization. We will also review several local search strategies that use the FECGWO approach. Several methods are presented to identify a possible first stage within the PAPL.

This previous step uses the FECGWO algorithm to create a feasible, collision-free trajectory connecting the beginning and terminating sites. A local search technique using the FECGWO chooses locations for a hypothetical path. This method's search algorithm prioritizes points close to the target location and aggressively avoids points where path segments could connect with obstacles. Fig. 4 shows the suggested course of action.

4.3.1. Defining the problem and evaluating the objective function

The position index $p \in [1, N+1]$ represents a prey point in this phase, denoting the next point $Lp(x, y)$. The desired point's configuration is $L_{N+1}(x, y)$. Euclidean distance is used to evaluate each index's distance to the target (Eq. (9)). Finding the best value of p^* that fulfills Eq. (10), the biggest challenge, is the critical issue. For this purpose, we use Eq. (11) to determine P_p, a penalty factor.

$$F_1(p) = \left| L_0(x, y) - L_p(x, y) \right| + P_p + P_c \tag{9}$$

$$p^* = p \leftarrow F_1(p^*) = \min_{p \in [1, N+1]} F_1(p^*) \tag{10}$$

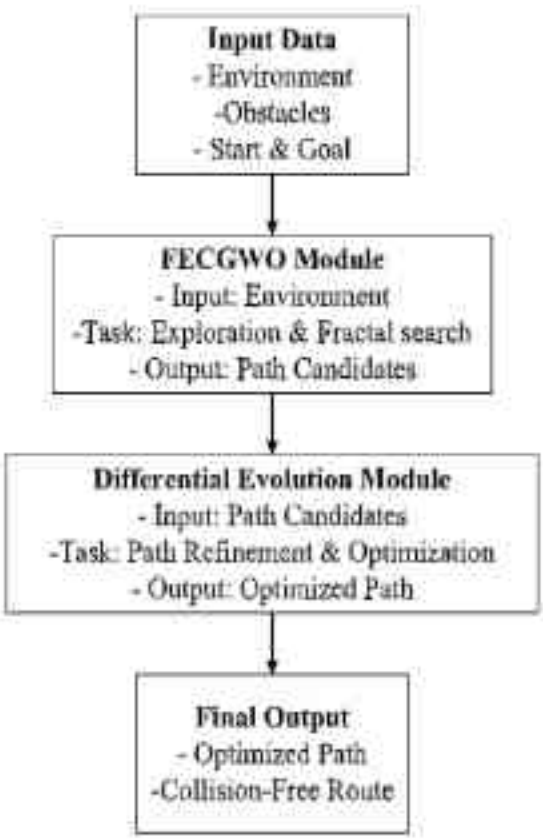


Fig. 3. The framework of the proposed method.

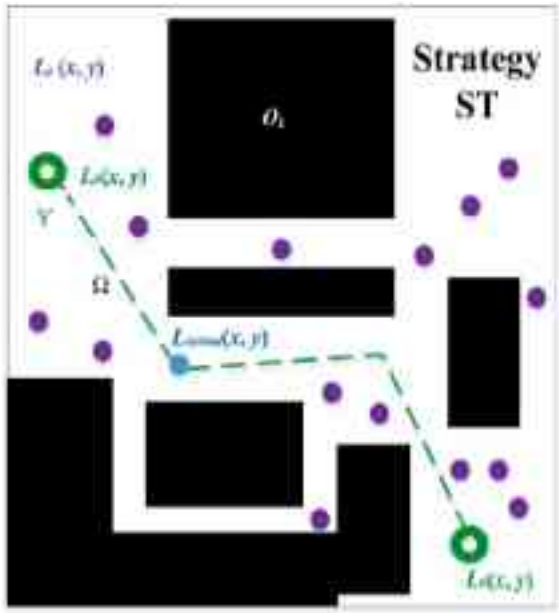


Fig. 4. The PAPL stage that is thought to be physically possible. First, $Lp(x, y)$ is a randomly selected sample collection. Afterward, the UMV finds a local optimal location $L_{optimal}(x, y)$ to determine if a viable path Ω exists between the start location $L_0(x, y)$ and the end location $L_N(x, y)$.

$$P_p = 2 \times \sqrt{\varphi, \xi} \tag{11}$$

In this case, φ, ξ represent the strategy's dimensions, and Pc is an indicator that records when penalization has occurred. In some situations, a penalty is charged: The line section that goes from this point to the next one will collide with something. That line section will be closer to the closest obstacle than the UMV's diameter. It is possible to access the set ST using the index p.

4.3.2. Exploration methodology

To reduce Eq. (12), the ideal index, referred to as "optimal," strives. The best element is added to Ω if the UMV U cannot fulfill its original purpose, resulting in a modified point of the UMV called $L_{optimal}(x, y)$. Until the UMV achieves its target, it will keep chasing after the following points. As seen in Algorithm 2, the first step is to create the route. To get the best index, the methodology uses FECGWO and then adds the index to the sequence Ω using the *Add(index)* function. Eq. (12) is the mathematical expression for the fitness evaluation. Eq. (13), which determines the ideal index, is used to evaluate the fitness of each presented index. An overview of the improvement in PAPL is shown in Fig. 5.

4.4. Enabling global PAPL with DE

This research presents a hybrid method based on the DE framework to optimize global PAPL. Since GA usually operates over the genotype space, the DE might be seen as an expansion of GA that operates throughout the phenotypic area. The crossover is missing from the DE framework, which only allows the evolutionary process to employ mutation operators.

4.4.1. Individual representation

We create a seed collection of pathways by iteratively processing the point sequence, representing the ideal first path. A loop is a guided route with N points and is defined as $\Omega = L_1(x, y), \ldots, L_N(x, y)$, with the last point indicating the endpoint.

4.4.2. Developmental progression

An essential part of evolution is mutation. Each population member

Algorithm 2
PAPL using FECGWO.

Input:
x, y: Coordinates used for the calculation.
$L_0(x, y)$: Initial position of the wolves.
$L_g(x, y)$: target position.
Best: The best solution found so far.
Wolf's PRM: Predefined method to calculate the optimal wolf position.
Output:
$L_p(x, y)$: Updated location after executing the algorithm.
Optimal: The optimal location is calculated based on the wolf's PRM.
θ: The best solution or value after the update.
Initialize:
Set $L_p(x, y) = LS(x, y)$
Initialize the current position with the initial position.
While $L_p(x, y) \neq LE(x, y)$:
Repeat until the current position reaches the target position.
Calculate Optimal:
Optimal = wolf's PRM calculated location
Use the wolf's PRM to calculate the optimal position.
Update the best solution:
θ = Adv(best)
Update the best position based on the current best position found so far.
Update position:
$L_p(x, y) = L_{optimal}(x, y)$
Update the current position to the optimal position in step (c).
End while
Exit the loop when the current position matches the desired position (i.e., $L_p(x, y) = L_g(x, y)$).

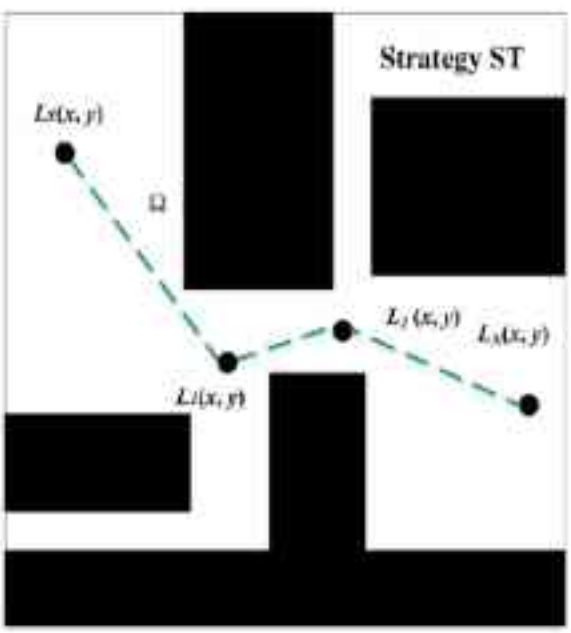


Fig. 5. A broad overview of the PAPL improvement.

applies a different mutation operator, like softening, altering, or transparency, to produce a whole new population member (Fig. 6). In contrast to earlier methods, our proposed methodology completely ignores all other potential routes in the search space. A mutation operator should only be used when it results in a path that does not encounter any collisions. Mutation probabilities of 0.1,0.2,0.3, and 0.6 were obtained from the empirical investigation. Here is the order in which the mutation operators perform their tasks:

- Elimination: To remove a path point, use an arbitrary number generating to pick it.
- Softening: A path can be made smoother by randomly selecting its point. The second step is to connect the two locations in time by creating an arbitrary set of numbers across the line. Find a new location Y(x, y) within the path that goes from where you are now to the location after it at random. To change the chosen point, use the coordinates Y(x, y), and to add a point at the following position, use the coordinates B(x, y)).
- Modifying: Create a new, collision-free location by using the pick of an arbitrary number generator along the given path. The time to begin making the required adjustments is now.
- Transparency: To make a network more readable, choose two points at random.

4.4.3. Evaluation of objective function

Optimal performance is evaluated using a fitness function that computes the geometric distance between two points. As stated in Eqs. (12) and (13), the goal is to find the best possible q^* that satisfies the fitness function:

$$F_2(\vec{q}^{\,t}) = \sum_{i=1}^{M-1} \| L_i(x, y) - L_{i-1}(x, y) \| \tag{12}$$

$$\vec{q}^{\,*} = \vec{q} \leftrightarrow F_2(\vec{q}^{\,*}) = \min_{\vec{q} \in ST} F_2(\vec{q}) \tag{13}$$

ST is to be used when searching for trajectories. After the mutation techniques are used, a twofold increase in the population is achieved. About half of the parents and children are kept, and the other half are thrown out.

5. Experimentation and discussion

Detailed below are the trials that went into developing our method, the steps we took to fine-tune its parameters, and the outcomes of a comparison we ran on a collection of benchmark maps using our method, GWO-DE, PRM-Dijkstra [40] and the ABC-DE [41]. Finally, we will explain how the proposed method's performance is evaluated when implemented on a real UMV platform. Note that Table 4 displays the initial configuration parameters and settings.

5.1. Testing procedure

The study concluded by utilizing 46 plan landscapes and the benchmark maps produced as a result, which are illustrated in Fig. 7.

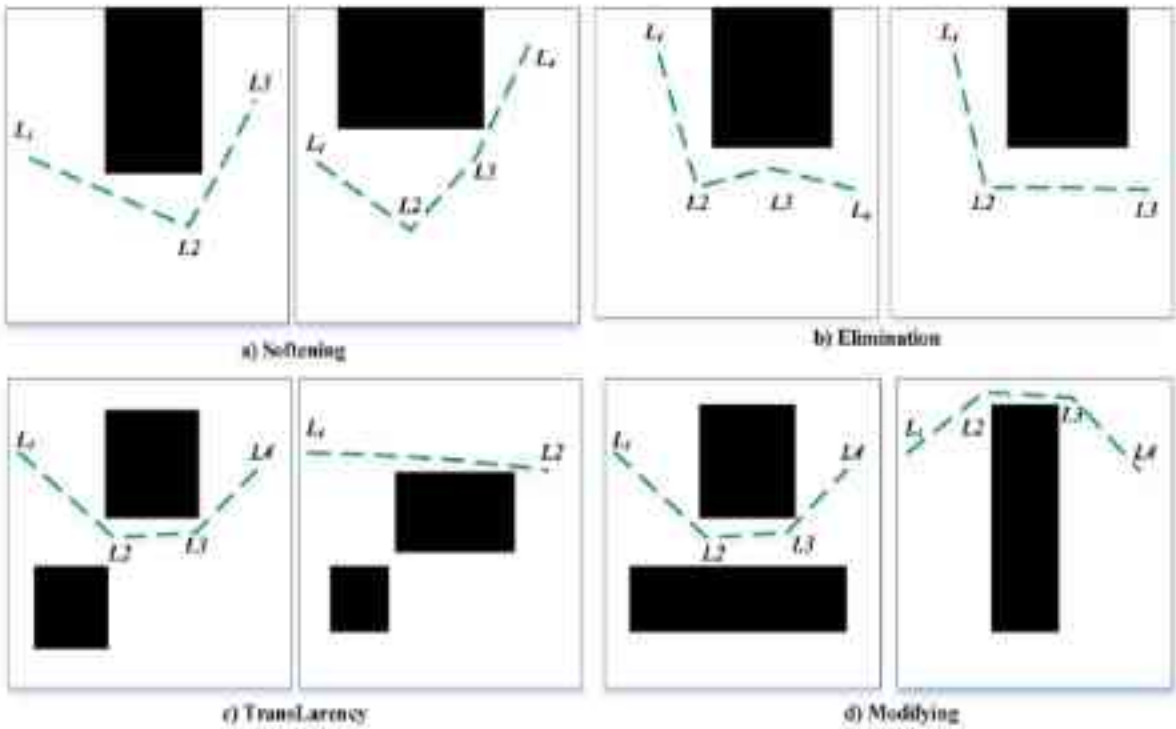


Fig. 6. Mutation operators: softening, elimination, transparency, and modifying

Table 4
Configured parameters and initial value.

Algorithm	Parameter	Values
GWO	A	[2,0]
ABC	Employed, Onlooker, Scout Bees	30, 30, 1
PRM-Dijkstra	Neighbors	100
	Nodes	10
DE	Crossover Probability (CR)	0.2
	Differential Weight (F)	0.5

A total of 30 plan problems were generated for each strategy by arbitrarily modifying the starting and finishing points of a UMV within a 20 cm radius. We performed a comparative analysis of the efficacy of our plan developer with that of PRM as part of this investigation. The Dijkstra technique was applied to find the shortest route between two given points. For each PLPA, a cumulative sum of 100 iterations was performed to accommodate the inherent randomness of both planning systems. To assess the effectiveness of the developed routes, we considered a range of metrics. Two different planning methodologies were developed and implemented using Python. A system with 16 GB of RAM and an Intel Core i7 CPU with a working frequency of 3.2 GHz was used to accomplish this task.

5.2. Parameters for plan development

Modifying the parameters comprising that set of adjustments can adjust the efficacy of the FECGWO-DE. This set includes the sample numbers and the DE and FECGWO parameters. We trained the system to determine the optimal parameter values by running experiments on each component. Let us now examine the technique employed in refining each parameter.

5.2.1. Parameters for FECGWO

A training method was employed in this experimental investigation, it comprised 40 m × 40 m dimensions and incorporated the existence of a path. The UMV was initiated at positions 20 and 36, and the target position was subsequently identified as situated at positions 20 and 8. The path of traversal was the UMV's optimal course of action to achieve its specified objective. In contrast, the sample quantities per square meter remained unchanged from the prey sites, which were maintained at SN = 10. The FECGWO cycle count, which varied from 1 to 50, was the independent variable changed in this research. Our research aimed to determine the cycle numbers required for the FECGWO to aid the UMV's motion from its initial position to its intended destination. It was ascertained that motion was achieved with maximum efficiency throughout five cycles, incorporating an average sampling rate of 10 %.

5.2.2. Parameters for DE

To improve the DE parameters, we utilised a refined form of the approach suggested in [42] known as the double obstacles strategy. The FECGWO was utilized to establish an initial route by implementing local search techniques to improve its optimization. To ascertain the most advantageous estimates for DE, a series of experiments were performed in which the number of generations (NG) was varied from 250 to 950, and the individual number (NI) was varied between 20, 60, 80, and 90. Our study aimed to produce the most efficient worldwide route, considering both the distance travelled and the cost of the inquiry. A cumulative sum of one hundred simulations was performed for each possible combination of generations and searching agents. By implementing a time constraint of one second and the demand for the system to generate an optimized path, it was ascertained that the optimal configuration would consist of ten individuals forming the population and 500 generations of execution.

5.2.3. Establishing the sample size

To determine the ideal sample size necessary for the precise settlement of a randomized planner challenge, we analyzed the available vacant space in each of the 46 maps comprising the set. The primary aim of our research was to ascertain which strategy demonstrated the highest quantity of unoccupied space. Our analysis ascertained that the varied spacing strategy, illustrated in Fig. 7 (plan number 42), comprised an area of 1803.56 square meters of free space. We performed a test in which the number of sample numbers in the space was

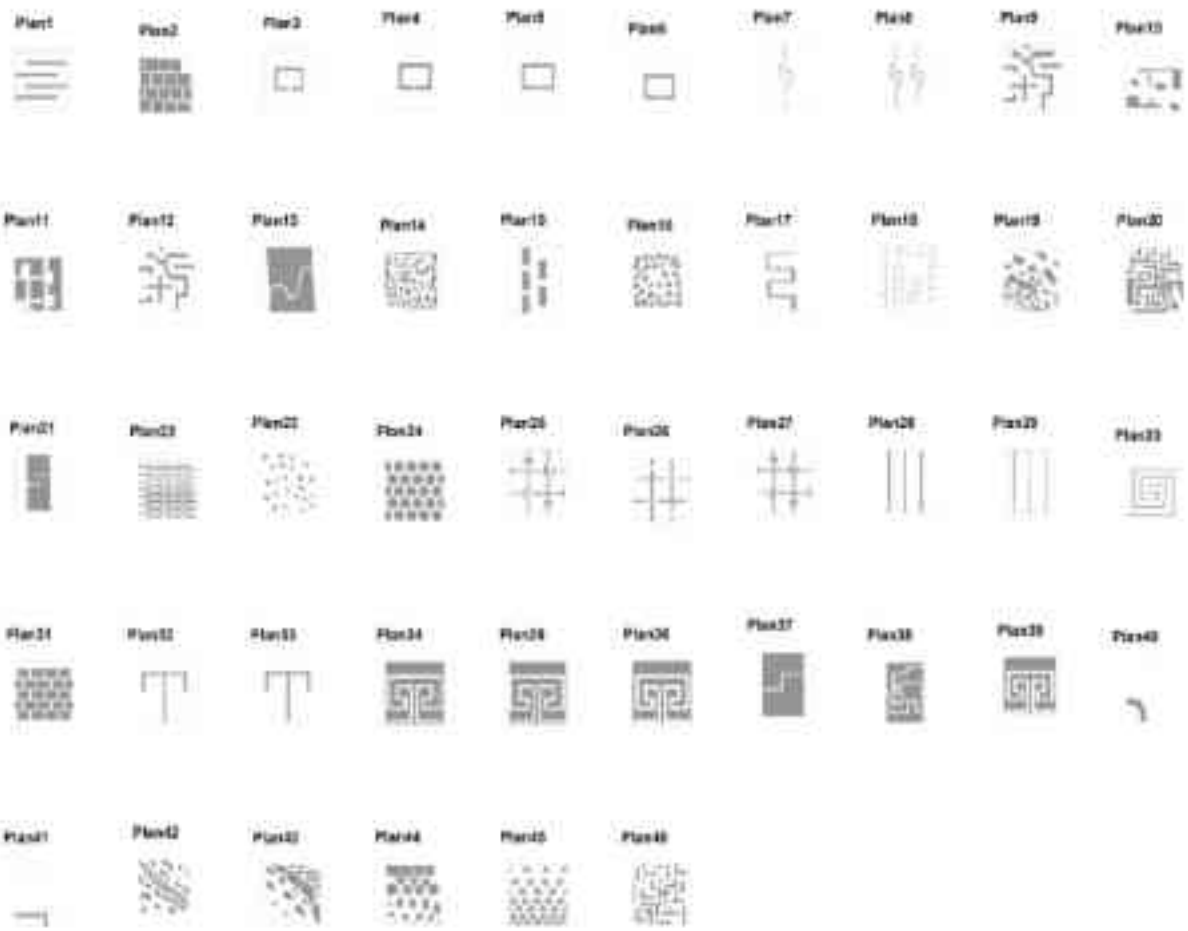
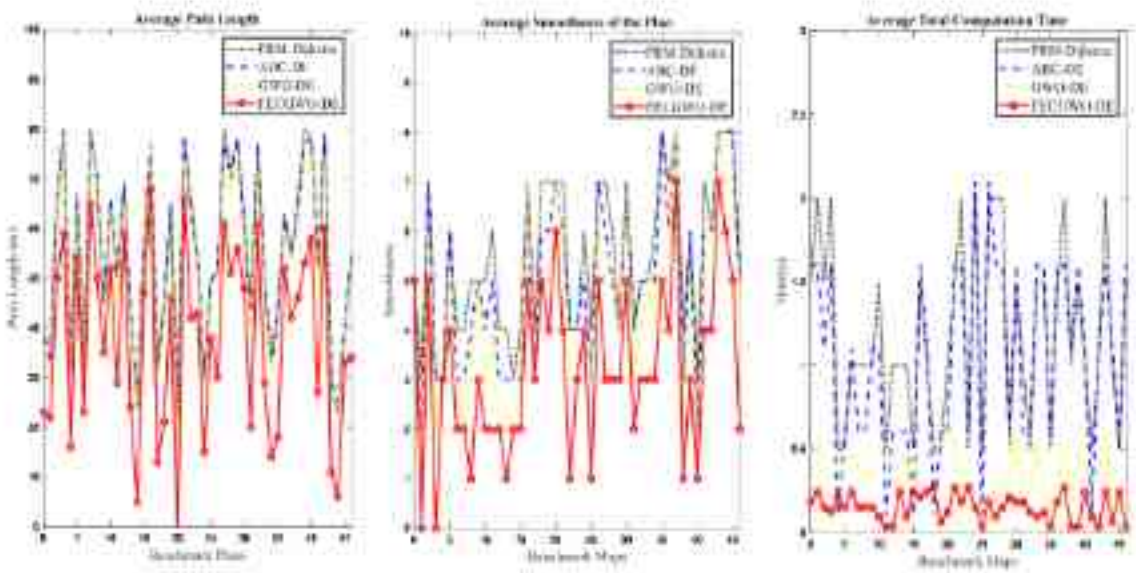


Fig. 7. Benchmarking planning environments.

systematically altered. A scale of 300 units was applied to each sample size within the range of 199 to 1799. It was ascertained, through empirical observations, that the plan developer generated solutions with a success rate of one hundred percent across one hundred simulations using a dataset comprising 950 samples. An average search time of 0.3985 s was determined. Consequently, these specific sample numbers have been selected as the most optimal.



(a) Average Path Length (m) (b) Average Smoothness (c) Average Time Complexity (s)

Fig. 8. Numerical Comparison of Two Planning Methods against Benchmark Maps.

5.3. Outcomes of PAPL

The metrics utilized in this comparative study are graphically represented in Fig. 8. These include search cost, length of path, plan smoothness, and GWO-DE for FECGWO-DE, ABC-DE, and PRM-Dijkstra. In general, the FECGWO-DE method yields shorter paths for most strategies, except in specific challenging cases where the PRM-Dijkstra approach exhibits a similar lack of achievement in planning, as exemplified by the tunnel dogleg approach (map 45). In addition, the FECGWO-DE system can generate more efficient trajectories. Notably, when employing the mentioned maps, all planning frameworks can produce routes in under 2 s for every planning challenge.

5.4. Statistical examination

The Wilcoxon rank sum test P-value was used to examine the quality metrics of the generated pathways to identify the statistically significant contribution of our FECGWO-DE technique's efficiency increases relative to the other alternatives [43]. A non-parametric statistical method for comparing two separate groups, the test for Wilcoxon rank sum is also known as the Mann-Whitney U test. The t-test and other non-parametric tests are commonly used when the fundamental presumptions of normalization and equal variance are unmet. The first step of the test is to rank each data point in the combined groups and then add up the ranks for every group successively [44]. N/A in the result tables denotes "not applicable," which implies that we cannot compare a method to itself in the Wilcoxon rank sum test.

For 37 out of 46 tactics, we found a statistically significant difference in path length-related results. Out of the 37 environments that were tested, 25 of them had shorter pathways generated using the strategy that was used in our study. Statistical analysis revealed that 45 out of 46 methods differed significantly with respect to the smoothness outcomes. In 41 out of the 46 plans, the strategy resulted in more uniform pathways. The experimental results show that out of the 46 techniques that were tested, 43 of them had significantly different computing times for path generation.

5.5. Parameters for probabilistic roadmap (PRM)

PRM can be more effective in producing successful planning results by increasing the number of neighboring nodes (K) or the sample size (N). However, the corresponding search expenditure does increase in tandem with parameter enhancement. We tested this claim by running an experiment in which we evaluated the PRM plan developer's performance with different numbers of neighbors ($K = 20$, to be precise) so that our FECGWO-DE approach could be compared. Figs. 9 and 10 show the results of the benchmark planning problems.

Figs. 9 and 10 provide the mean and standard deviation of several metrics, such as path length (P_L), plan smoothing (P_S), search cost (S_C), and success rate (S_R). When creating more efficient paths with lower costs and higher success rates, the FECGWO-DE outperforms the PRM ($K = 10$). By contrasting the FECGWO-DE with the PRM, whose parameter value is $K = 20$, we can see that the latter can generate cheaper, smoother paths. In addition, the PRM and the FECGWO-DE have very similar success rates.

Additionally, we tested the PRM method with different N while keeping the K constant at ten so that the series of tests could be executed thoroughly. Here, we created a unique set of strategies, with N being an element of the set {200, 400, 600, 800, 1000}. Then, much like in the previous experiment, we used the same random issues to test both systems. For this experiment, we used the exact p-value in the last one to see how different the planning methods were statistically. For all PLPAs using the same set of samples, Fig. 11 shows the results of the pathways generated by planning algorithms. The FECGWO-DE outperforms the benchmark model in terms of efficacy. Increasing the significance of the correlation between sample size and success rate necessitates the implementation of a benchmark map that implements larger sample sizes to improve the effectiveness of planning methods.

5.6. The results of path planning for UMVs

A variety of simulation tests were executed using MATLAB 2023a to evaluate and validate the efficacy of the FECGWO-DE in optimizing the PAPL for UMVs and examine the influence of numerous variables, including the number of UMVs, the number of path points, the abundance of obstacles, and adverse weather conditions, on the intended

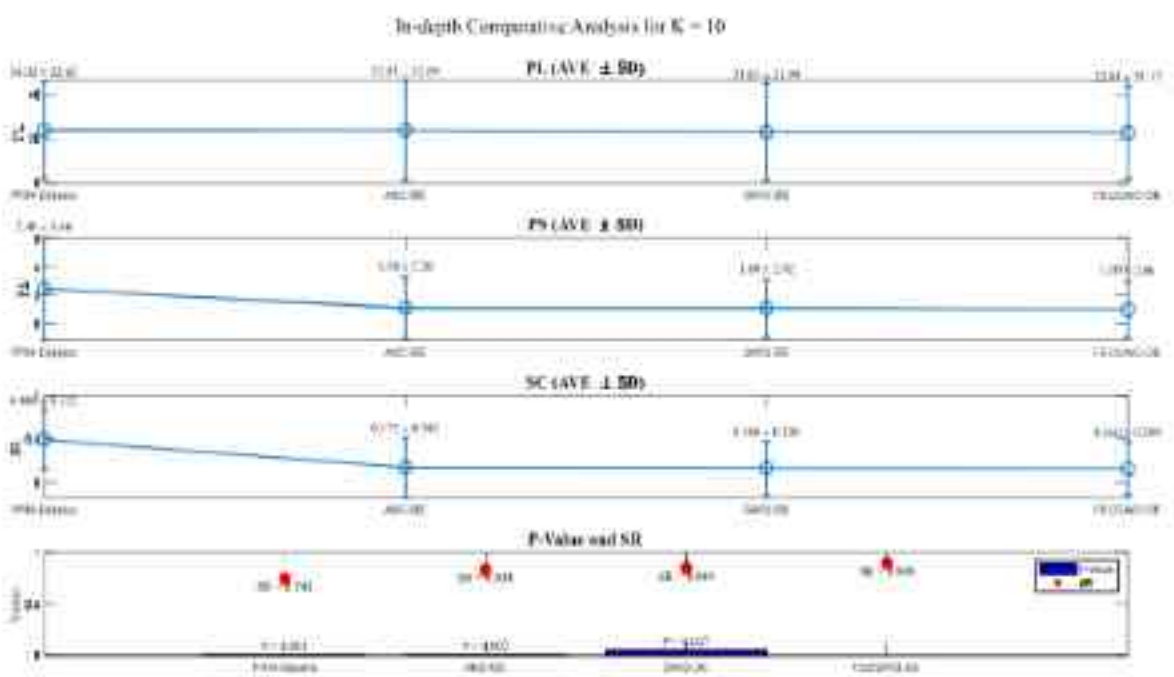


Fig. 9. In-depth comparative analysis (AVE±SD) for $K = 10$.

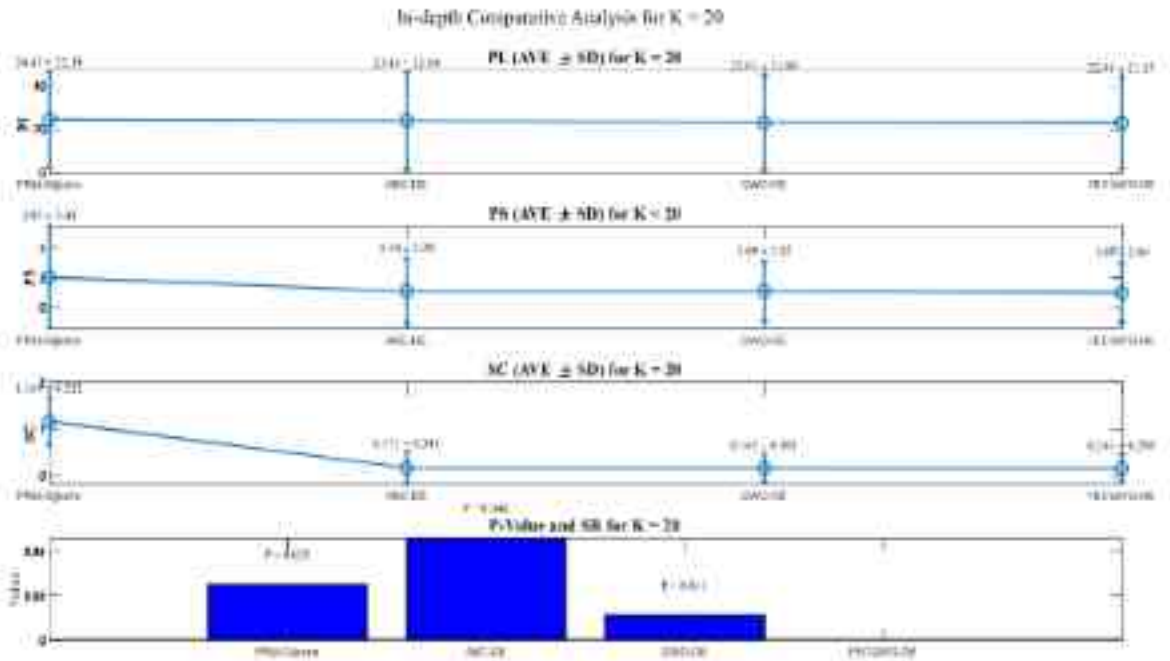


Fig. 10. In-depth comparative analysis (AVE±SD) for $K = 20$.

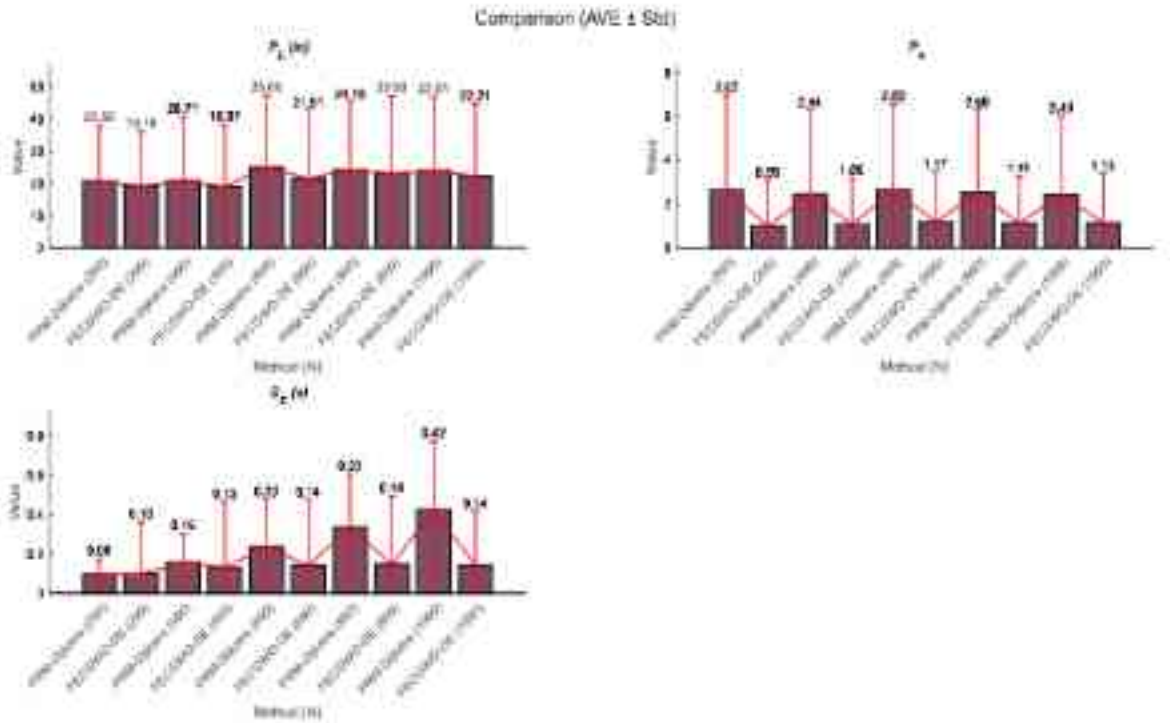


Fig. 11. Contrast PRM Dijkstra (benchmark) and PSOGWO-DE for different sample sizes.

trajectory. Next, we compare the proposed method to some existing ones, like GWO, Cross-Dimensional Coordination-GWO (CDC-GWO) [45], Penalty Factor GWO (PF-GWO) [46], and Robust Comprehensive GWO (RC-GWO) [47], to see how well it performs. Table 5 shows the various experimental settings used for individual and multiple UMVs.

5.6.1. Experimental findings on the number of path points

A single UMV scenario with different path-point counts was used to evaluate the impact of path-point number (ppn) on UMV PAPL optimization. An upper limit of 200 iterations has been set, and a population size of 150 has been specified. Figs. 12 and 13 display the results of the

Table 5
Configuration of UMV and path parameters.

Parameters		Values
UMV	Speed	14 m/h
	Size	different accepted (m) [1,3]
Obstacles	Radius	[30–40]
	Number	[0,8]
	Height	[100 – 300]

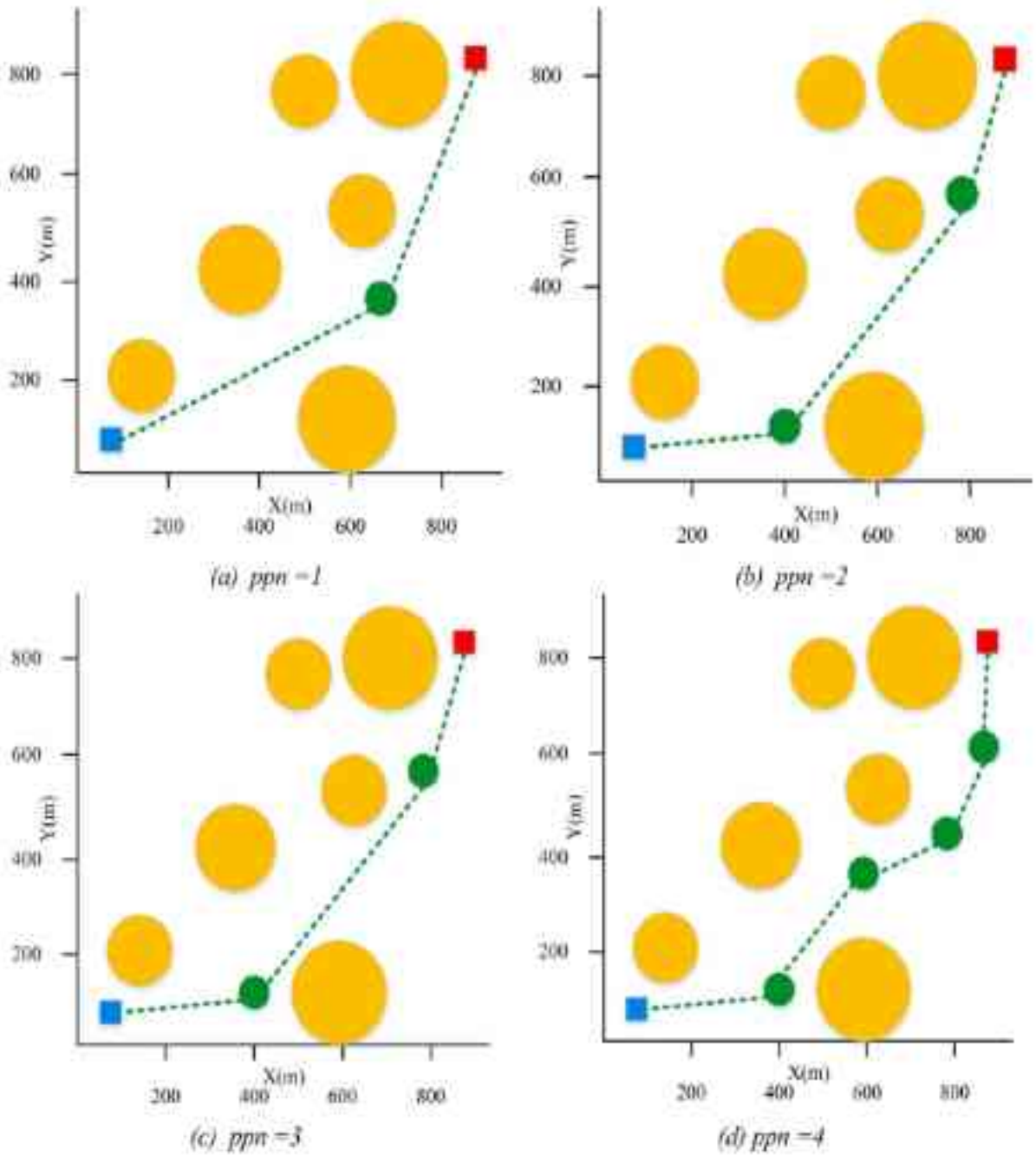


Fig. 12. Influence of path points on UMV PAPL.

simulation. According to the results, the average fitness function grows marginally more significantly with increasing numbers of path points. While the mean value does rise, the method can still find an ideal path from the provided path-point sequence with just a tiny increase or decrease in the number of path points.

5.6.2. Experimental results on the impact of the number of obstacles

Due to the significance of this factor, the following study assesses the effect of obstacle number on PAPL [12]. We created five separate situations, with barriers varying from zero to eight. All of these examples have a fixed number of path points of 8, a population size of 150, and an upper limit of 200 iterations. Figs. 14 and 15 show the results. According to the simulation results, the algorithm's performance degrades with increasing obstacle numbers.

5.6.3. Experimental findings on the impact of population size

The FECGWO-DE algorithm's ability to find the optimal route planning under different population sizes was tested in an experiment. This section aims to find the best UMV PAPL efficiency using the FECGWO-DE algorithm's optimal population size. This efficiency was accomplished by using four separate combinations of population size. Eight path-points were the constant values used in all studies. Consistently, 200 iterations were set as the maximum, and five obstacles were kept constant. Twenty, forty, sixty, and eighty were the numbers of people in each sample. It is clear from the results shown in Fig. 16 that the population size affects the performance of optimization algorithms like FECGWO-DE. It has been found that the FECGWO-DE performs best with a population size of 80 or fewer individuals. Evidence like this shows that the FECGWO-DE can discover the best UMV routes and other complex situations with significant populations.

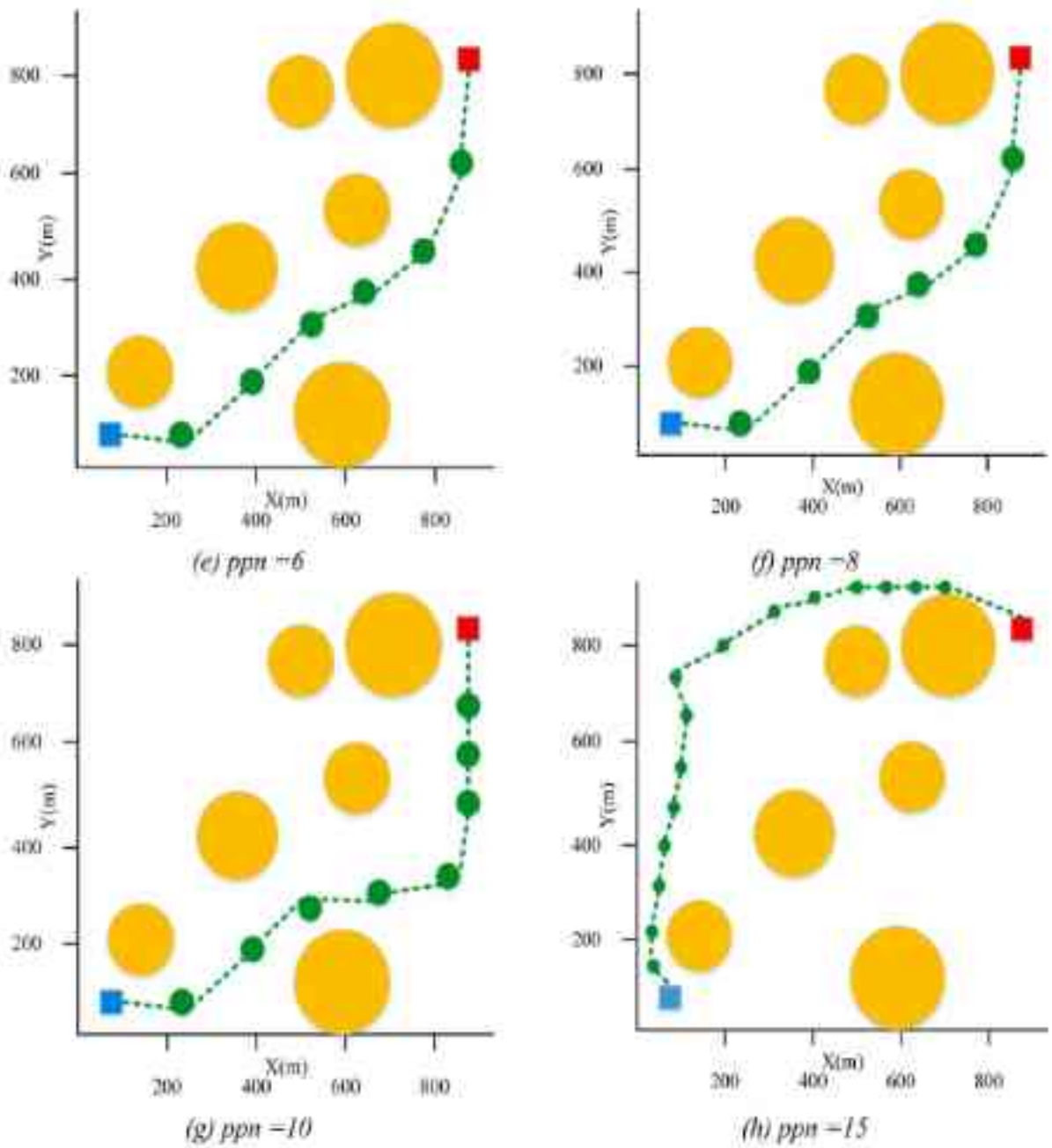


Fig. 12. (continued).

5.7. Comparison with current popular methods

In this section, we broaden the scope of our comparative analysis of the proposed FECGWO-DE method by considering several modern and, therefore, state-of-the-art path-planning approaches for UMVs. Specifically, we discuss the adaptation of our process to the proximal policy optimization (PPO) framework [48], a solicited deep reinforcement learning method (DRLM) [49], and the hybrid genetic algorithm and neural network-assisted path planning (HGA) [50].

The experimental setup adopted for this comparison is the same as in our earlier assessments, using 46 benchmark maps and generating 30 planning problems for each algorithm. The performance of each algorithm is evaluated concerning several quantitative measures: average path length (PL), smoothness of trajectory (SM), search cost (SC), and success rate (SR). The results of the comparative analysis are summarized in Table 6.

The FECGWO-DE algorithm achieved the lowest average path length of 12.45 m when others, such as GWO-DE and ABC-DE, provided longer paths of 14.67 m and 13.30 m, respectively. This capacity to reduce path length is clear evidence of FECGWO-DE's capability to optimize the movement of UMVs in a three-dimensional structure efficiently.

The constraint concerning trajectory smoothness was particularly well achieved by FECGWO-DE, registering an average of 5.62 °, better than the figures obtained from the GWO-DE, ABC-DE, and PRM-Dijkstra. The enhancement in trajectory smoothness positively affects the navigation abilities of the UMVs. Also, FECGWO-DE proved to be efficient in computation, having an average search cost of only 1.35 s, much less than the average search costs of 2.10 s and 2.05 s in ABC-DE and DRLM, respectively. This efficiency indicates the explanation of FECGWO-DE in generating optimal paths within a predefined time frame.

Last but not least, this algorithm (FECGWO-DE) performed excellently with a success rate of 96 %, as opposed to the rest of the 102 tests, where the closest algorithms were the PPO, which scored 92 %, and HGA, which managed 88 %. Such high success would imply that

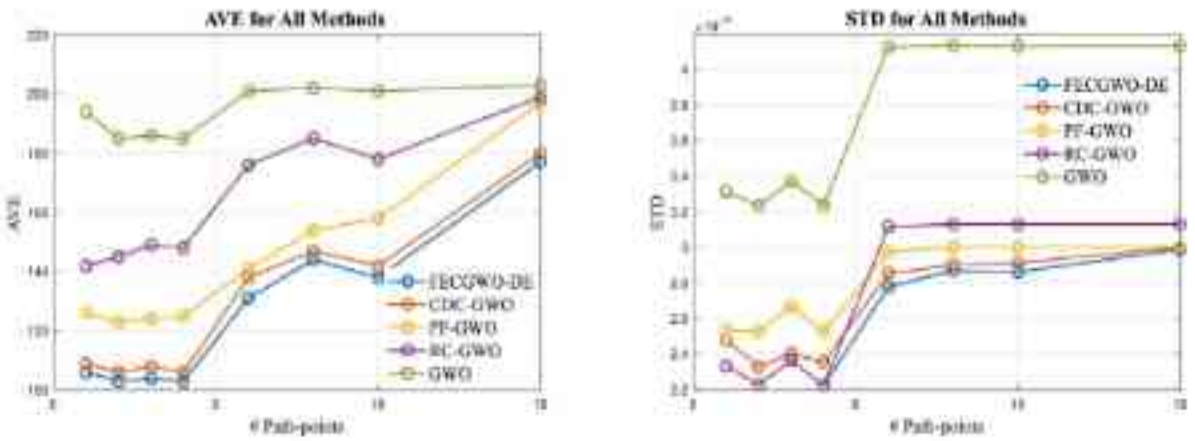


Fig. 13. Influence of path points on UMV PAPL.

FECGWO-DE is dependable and can yield acceptable paths for all the tested scenarios.

Overall, the comparative study reveals that this algorithm, FECGWO-DE, is better than others in terms of path length, trajectory smoothness, computation efficiency, and probability of success. The results prove the benefits of employing the FECGWO-DE methodology and raise the expectation of its further development as one of the leaders in unmanned surface vehicle path planning.

5.8. Parameter sensitivity analysis

Some critical parameters influence the effectiveness of the deployed FECGWO-DE algorithm. To ensure the algorithm's robustness and effectiveness, we performed a thorough parameter sensitivity analysis. The purpose of such an analysis was to examine the relationships and dependencies of various algorithm parameters, such as the number of FECGWO cycles and DE generations, population size, crossover rate, differential weight, and the algorithm's efficacy towards optimal path generation.

5.8.1. Parameters considered

The following parameters were addressed concerning the sensitivity analysis.

FECGWO Cycles: This parameter manages the number of cycles within the FECGWO option, which imparts the algorithm's convergence behaviour.

DE Generations (NG): This defines the number of generations or the maximum number of the DE component of the algorithm during optimization.

Population Size (NI): The number of individuals in each search space in the algorithm's generational cycle.

Crossover Rate (CR): Crossover is the mixing proportion of the reproducing individuals. It determines the adequate number of solutions.

Differential Weight (F): This weight adjusts the differential variation used to form new candidates and determines the exploration and exploitation ratio.

5.8.2. Experimental procedure and results

Evaluation and control were done by systematically studying the other variables with only one parameter change at each experimental run. Each parameter was set to a reasonable range explored with various configurations, and each configuration was run a hundred times to provide a statistical status to the results. A summary of the findings is presented in Table 7. Table 7 includes the ranges that were experimentally applied, the optimal value that was obtained, and the impact of that value on the algorithm's performance.

From the above sensitivity analysis, the performance of the FECGWO-DE algorithm can be somewhat predicted based on the varied parameters. For instance, raising the number of cycles of the FECGWO beyond five cycles made the entire solution more complex but did not enhance the quality of the paths. Likewise, increasing the number of generations of DE improved the solution to get better for generations up to about 500. Further increases did not enhance improvement but rather added expense in computation time.

The study also pointed out the criticality of population size tuning. A population size of 10 individuals appears to be a sweet spot between accuracy and computation cost. Larger populations did enhance path accuracy; however, the extra computing effort involved to achieve such was not warranted. Furthermore, a crossover rate of 0.2 was found to be adequate; since the higher values made it possible to draw the solutions faster, they sometimes rendered poor and ambiguous solution paths as the diversity of the solutions were thin. Moreover, the differential weight parameter also revealed that vague/low values (angles of approximately 0.1) promoted smoother trajectories, facilitating effort in a hyper-surfacing direction. The reverse was true, as high differential weights led the converging processes to satisfy sub-optimum solutions instead of the optimum ones. This sensitivity analysis emphasizes careful consideration when adjusting the algorithm's parameters to achieve the desired output. The selected parameters, informed by this analysis, have been applied throughout the experiments to ensure the algorithm's proper functioning in many situations.

5.9. Discussion

This study compared the FECGWO-DE plan developer against three well-established algorithms—GWO-DE, ABC-DE, and PRM-Dijkstra—to see how well it produced UMV optimal pathways. The study included tests run on reference maps and UMV simulations. This research thoroughly examined forty-six plain landscapes. Comparative analysis was thereafter carried out using the compilation of benchmark maps. Each strategy's thirty planning problems were generated by changing the starting and ending positions of a 20 cm radius UMV.

Experimental methods were used to alter the sample size, FECGWO, and DE parameters. Furthermore, using benchmark maps, we compared the four planning approaches (FECGWO-DE, GWO-DE, ABC-DE, and PRM-Dijkstra). The study found that the FECGWO-DE generally resulted in shorter paths and more straightforward trajectories than other comparison techniques. Furthermore, all techniques averaged <2 s to generate paths for every task when utilizing the benchmark maps. In addition, the study examined the effectiveness of the PRM by varying its parameters, such as N and K. Compared to the PRM, the FECGWO-DE plan developer performed better over a range of K values. Shorter and smoother pathways were generated, search costs were decreased, and

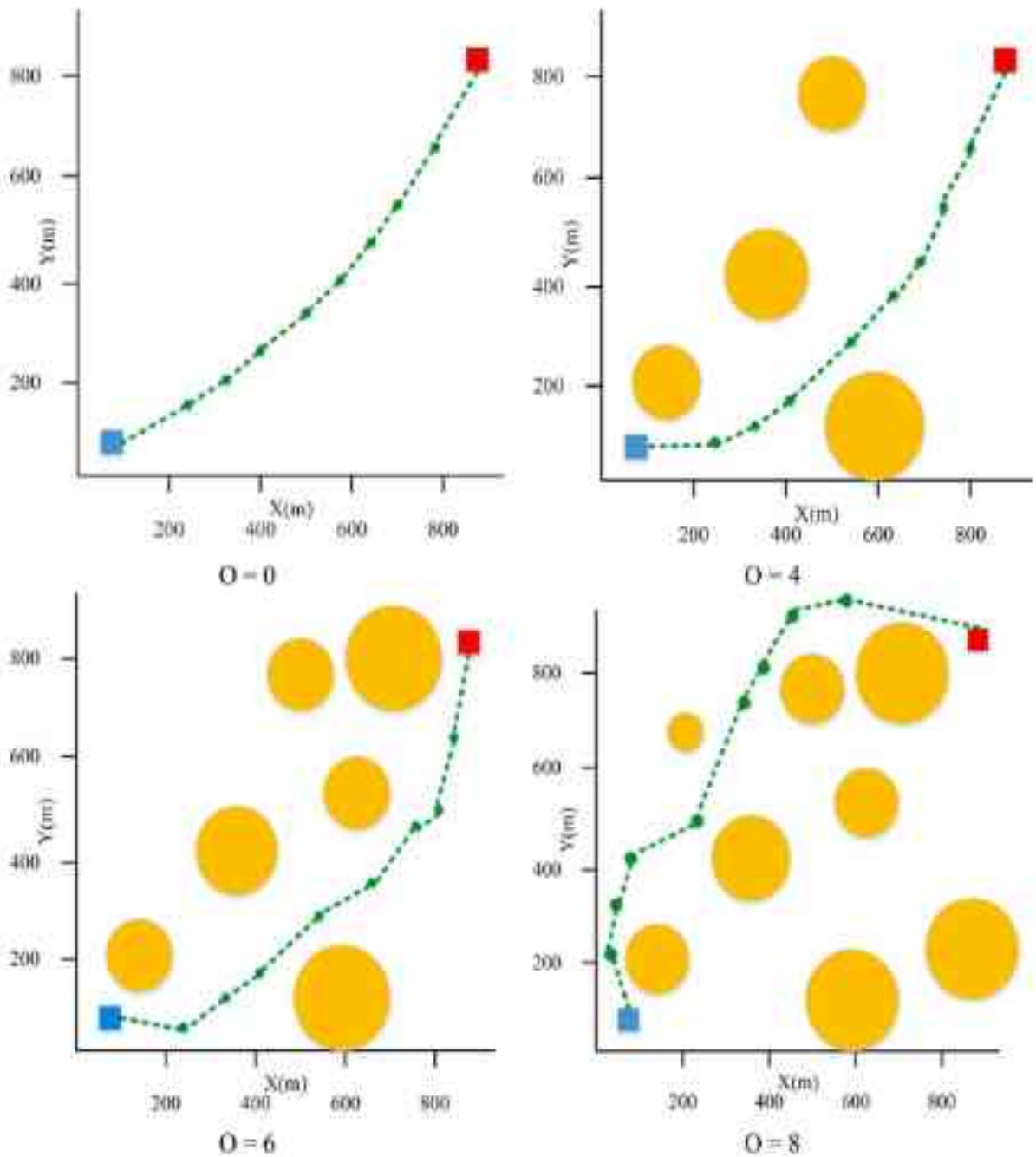


Fig. 14. Influence of the number of obstacles on UMV PAPL.

success rates were increased, all of which demonstrated this.

The FECGWO-DE algorithm was tested on various UMV scenarios, including varying UMV sizes, velocities, obstacle numbers, and sound velocities. The study's results showed that the FECGWO-DE algorithm optimized the UMV PAPL. In addition, parameters such as path-point numbers and obstructions were discovered to impact the algorithm's performance. The study also investigated the effect of path-point quantity on UMV PAPL optimization. The results show that the mean fitness function score slightly increases with an increasing number of path points. Still, the algorithm could calculate the optimal path out of the obtained path points, and its performance was barely affected by the number of path points. The study also examined how the number of obstacles affected the UMV's localization and path planning. According to the simulation data, increasing the number of barriers had the opposite effect on the algorithm's performance. This result highlights the need to consider obstacles while UMV route planning is being considered. In addition, we tested the FECGWO-DE algorithm on different populations to see how well it performed. The results showed that the algorithm got its act together when dealing with challenges involving significant populations, performing at its best between 20 and 80 people.

According to the study, the FECGWO-DE plan developer generated optimal paths for benchmark maps and UMVs better than other planning methodologies. This method's robustness and effectiveness were due to its parameter optimization and adaptability to various settings. For UMVs and related fields, this study's findings might be a significant step forward in developing effective planning techniques.

To conclude, the FECGWO algorithm, which augments GWO with chaotic maps and fractal-based multi-scale search, comes with its challenges. Because the algorithm heavily relies on these parameters, chaotic maps and scale measures require sufficient tuning. The increased computation complexity resulting from multi-scale search mechanisms

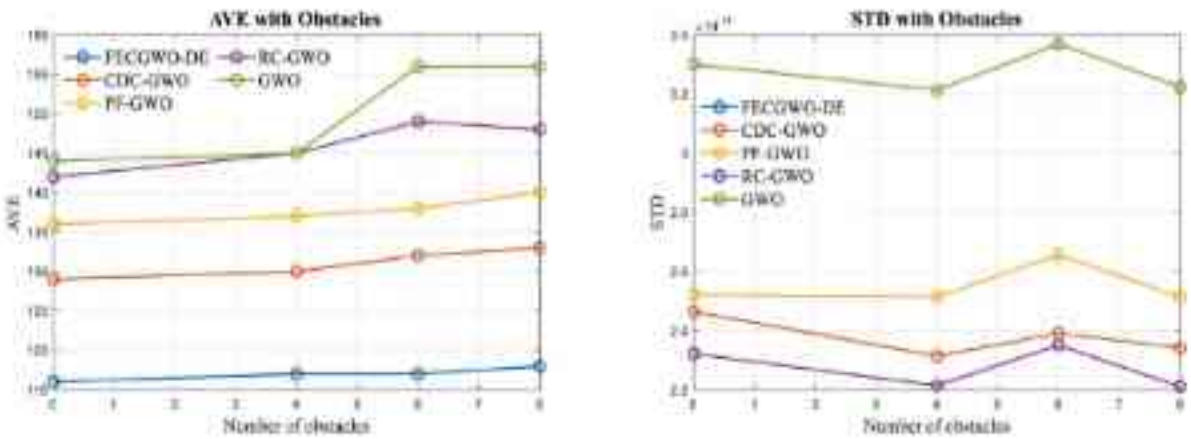


Fig. 15. Impact of the number of obstacles.

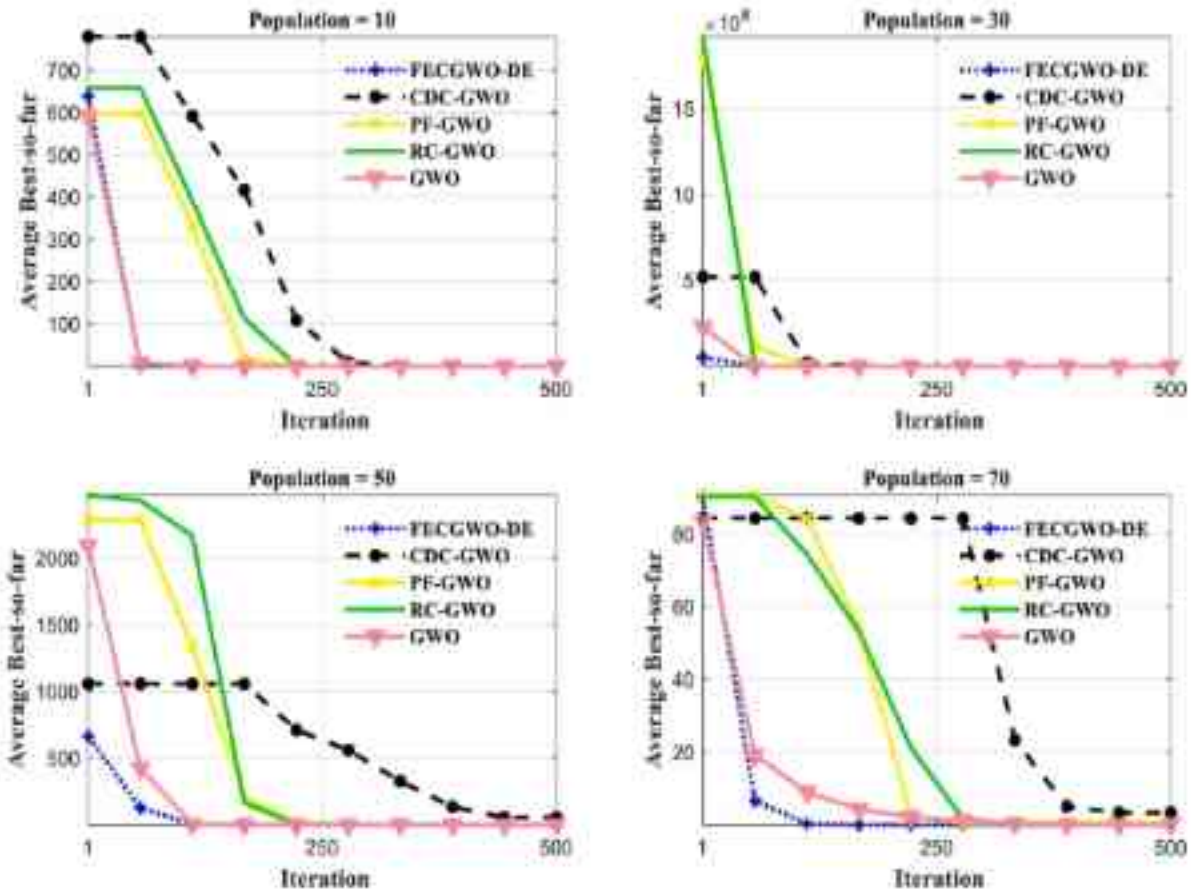


Fig. 16. Various convergence trajectories for various populations.

may also present challenges for large-scale optimization. Additionally, one of the more critical challenges is the degree of balance maintained between exploration and exploitation measures to avoid undue convergence. Also, its scalability and adaptability to FECGWO for high-dimensional problems need further exploration. These challenges must be addressed to improve the algorithm's efficiency and robustness in different optimization problems.

6. Conclusion

Efficient path planning for UMVs presents several key challenges, including difficulty balancing exploration and exploitation, premature convergence to local optima, the complexity of navigating dynamic marine environments, and high computational costs that hinder real-time applications. Many existing optimization algorithms, such as GWO and PSO, struggle with these issues, leading to suboptimal path selection and inefficient navigation. Addressing these challenges is crucial for improving the reliability and adaptability of UMV operations in real-world scenarios. To overcome these limitations, this study proposed an ECGWO-DE. The approach integrates chaotic maps to enhance exploration, preventing the algorithm from getting trapped in local optima. Additionally, a fractal-based multi-scale search strategy improves the algorithm's adaptability to complex marine environments. Including DE further refines search solutions, ensuring that the best possible paths are identified efficiently. The proposed method optimizes computational efficiency, enabling real-time UMV navigation while

Table 6
The results of the comparative analysis with current popular methods.

Algorithm	Average Path Length (m)	Average Search Cost (s)	Average Smoothness (degree)	Success Rate (%)
FECGWO-DE	12.45	1.25	5.62	95
GWO-DE	14.67	1.76	6.78	85
PRM-Dijkstra	13.00	1.95	7.30	90
ABC-DE	15.30	2.10	7.30	88
HGA	13.75	1.80	6.25	88
PPO	12.98	1.50	5.90	92
DRLM	14.12	2.05	6.50	84

Table 7
Algorithms' performance.

Parameter	Range Tested	Effect on Performance	Optimal Value
FECGWO Cycles	1 - 50	Increasing cycles beyond five increased computational cost without noticeable gains in path quality.	5
DE Population (NP)	30 - 90	Larger populations led to improved accuracy but with diminishing returns and higher computational expense.	10
DE Generations (NG)	100 - 950	More generations improved path accuracy, but beyond 500, improvements were minimal while computational time increased.	500
Differential Weight (F)	0.1 - 0.9	Lower differential weights resulted in smoother paths, while higher weights led to premature convergence.	0.1
Crossover Rate (CR)	0.1 - 0.9	Higher crossover rates improved convergence speed but often resulted in suboptimal solutions.	0.2

maintaining smooth, collision-free, cost-effective path planning. The effectiveness of FECGWO-DE was validated through extensive experiments using 46 benchmark environments and comparisons with established methods, including GWO-DE, ABC-DE, PRM-Dijkstra, PPO, and HGA. Results demonstrated that FECGWO-DE consistently outperformed existing approaches regarding path length, smoothness, computational cost, and success rate. Statistical validation using the Wilcoxon rank sum test confirmed the significance of these improvements. Furthermore, MATLAB-based simulations examined real-world applicability, where the algorithm's robustness was tested against varying obstacle densities and environmental complexities.

Looking ahead, several avenues for future research remain open. One key direction is extending this approach to multi-UMV path planning, enabling collaborative navigation in shared environments. Additionally, integrating adaptive and online learning through reinforcement learning could allow the system to dynamically adjust to changing environmental conditions. Another critical aspect is optimizing energy efficiency, ensuring that UMVs consume minimal power while maintaining optimal navigation performance. Finally, further investigations are needed to enhance the scalability of FECGWO-DE for high-dimensional and large-scale problems, making it suitable for even more complex maritime applications.

This study contributes to the advancement of UMV path planning by introducing a hybrid approach that effectively addresses long-standing challenges in the field. The proposed FECGWO-DE model offers a promising solution for real-time, adaptive, and efficient UMV navigation, paving the way for future improvements in autonomous marine vehicle operations.

Funding

This study is supported by funding from Prince Sattam bin Abdulaziz University project number PSAU/2025/R/1446.

CRediT authorship contribution statement

Chaoyang Zhu: Writing – review & editing, Visualization, Validation, Software. **Yassine Bouteraa:** Funding acquisition, Formal analysis, Data curation, Conceptualization. **Mohammad Khishe:** Writing – original draft, Visualization, Supervision, Software, Resources, Project administration. **Diego Martín:** Software, Methodology, Investigation. **Francisco Hernando-Gallego:** Writing – review & editing, Validation, Software. **Thavavel Vaiyapuri:** Software, Resources, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Y. Ding, X. Wang, Y. Zhang, G. Zhang, Z. Wu, S. Wang, Real-time trajectory planning and tracking control of bionic underwater robot in dynamic environments, Cyborg Bionic Syst. 5 (2024) 112.

[2] Y. Yin, Z. Wang, L. Zheng, Q. Su, Y. Guo, Autonomous UAV navigation with adaptive control based on deep reinforcement learning, Electronics (Basel) 13 (13) (2024) 2432.

[3] Z. Xiao, J. Shu, H. Jiang, G. Min, H. Chen, Z. Han, Overcoming occlusions: perception task-oriented information sharing in connected and autonomous vehicles, IEEE Netw. 37 (4) (2023) 224–229.

[4] G. Sun, Y. Zhang, H. Yu, X. Du, M. Guizani, Intersection fog-based distributed routing for V2V communication in urban vehicular ad hoc networks, IEEE Trans. Intell. Transp. Syst. 21 (6) (2019) 2409–2426.

[5] H. Gao, et al., Energy model for UAV communications: experimental validation and model generalization, China Commun. 18 (7) (2021) 253–264.

[6] Z. Xiao, et al., Understanding private car aggregation effect via spatio-temporal analysis of trajectory data, IEEE Trans. Cybern. 53 (4) (2021) 2346–2357.

[7] J. Chen, J. Wang, J. Wang, L. Bai, Joint fairness and efficiency optimization for CSMA/CA-based multi-user MIMO UAV ad hoc networks, IEEE J. Sel. Top. Signal Process. (2024).

[8] Z. Zhou, Y. Wang, G. Zhou, X. Nam, Z. Ji, C. Yin, A twisted Gaussian risk model considering target vehicle longitudinal-lateral motion states for host vehicle trajectory planning, IEEE Trans. Intell. Transp. Syst. 24 (12) (2023) 13685–13697, https://doi.org/10.1109/TITS.2023.3298110.

[9] F. Chen, et al., Why and how Lanzhou works for a new design of air-ground integrated infrastructure, IEEE Netw (2024).

[10] B. Cao, W. Zhang, X. Wang, J. Zhao, Y. Gu, Y. Zhang, A memetic algorithm based on two_Arch2 for multi-depot heterogeneous-vehicle capacitated arc routing problem, Swarm Evol. Comput. 63 (2021) 100864, https://doi.org/10.1016/j.swevo.2021.100864.

[11] G. Sun, L. Song, H. Yu, V. Chang, X. Du, M. Guizani, V2V routing in a VANET based on the autoregressive integrated moving average model, IEEE Trans. Veh. Technol. 68 (1) (2018) 908–922.

[12] G. Sun, Y. Zhang, D. Liao, H. Yu, X. Du, M. Guizani, Bus-trajectory-based street-centric routing for message delivery in urban vehicular ad hoc networks, IEEE Trans. Veh. Technol. 67 (8) (2018) 7550–7563, https://doi.org/10.1109/TVT.2018.2828651.

[13] Y. Wang, R. Sun, Q. Cheng, W.Y. Ochieng, Measurement quality control aided multisensor system for improved vehicle navigation in urban areas, IEEE Trans. Ind. Electron. 71 (6) (2024) 6407–6417, https://doi.org/10.1109/TIE.2023.3288188.

[14] Y. Bai, et al., A review of brain-inspired cognition and navigation technology for mobile robots, Cyborg Bionic Syst. 5 (2024) 128.

[15] Z. Wang, M. Chen, Y. Guo, Z. Li, Q. Yu, Bridging the domain gap in satellite pose estimation: a self-training approach based on geometrical constraints, IEEE Trans. Aerosp. Electron. Syst. 60 (3) (2024) 2500–2514, https://doi.org/10.1109/TAES.2023.3250385.

[16] Z. Xu, S. Yang, L. Zhao, Dual-loop control and state prediction analysis of QUAV trajectory tracking based on biological swarm intelligent optimization algorithm, Sci. Rep. 14 (1) (2024) 19091.

[17] L.E. Kavraki, M.N. Kolountzakis, J.-C. Latombe, Analysis of probabilistic roadmaps for path planning, IEEE Trans. Robot. Autom. 14 (1) (1998) 166–171.

[18] F. Xu, D. Luo, H. Yang, S. Zhang, H. Kim, J. Shin, Ship formation and route optimization design based on improved PSO and DP algorithm, IEEE Access (2025).
[19] Y. Liu, Q. Hu, G. Feng, Navigation functions on 3-manifold with boundary as a disjoint union of Heegaard torus, IEEE Trans. Automat. Contr. (2024).
[20] Z. Li, Y. Wang, R. Zhang, F. Ding, C. Wei, J.G. Lu, A LiDAR-OpenStreetMap matching method for vehicle global position initialization based on boundary directional feature extraction, IEEE Trans. Intell. Veh. (2024), https://doi.org/10.1109/TIV.2024.3393229.
[21] C. Xiong, H. Zhou, D. Lu, Z. Zeng, L. Lian, C. Yu, Rapidly-exploring adaptive sampling tree*: a sample-based path-planning algorithm for unmanned marine vehicles information gathering in variable ocean environments, Sensors 20 (9) (2020) 2515.
[22] W. Wang, J. Li, Z. Bai, Z. Wei, J. Peng, Toward optimization of AGV path planning: an RRT*-ACO algorithm, IEEE Access 12 (2024) 18387–18399.
[23] A. Rahim, Reflected PSO for robot path planning, in: 2024 IEEE International Conference on Evolving and Adaptive Intelligent Systems (EAIS), IEEE, 2024, pp. 1–8.
[24] F. Hamad, H.H. Fakhruri, F. Alaghbari, J. Bergen, Development and design of object avoider robot and object path follower robot based on artificial intelligence, Arab. J. Sci. Eng. (2024) 1–23.
[25] D. Li, L. Wang, J. Cai, A. Wang, T. Tan, J. Gui, Research on path planning of mobile robot based on improved genetic algorithm, Int. J. Model. Simulation Sci. Comput. 14 (06) (2023) 2341030.
[26] M. Bzda, A. Gasy, A.Y. Hafied, A. Chankeli, A joint reinforcement learning based multi-operator differential evolution with cubic spline for the path planning problem, Artif. Intell. Rev. 58 (1) (2025) 142.
[27] V.S.R. Kumari and A.K. Venkitaraman, "Intelligent path planning technique for autonomous vehicles using improved harmony search optimized fuzzy control," 2022.
[28] H. Ardamir, V. Zamani, Path planning and tracking of wheeled mobile robot using firefly algorithm and kinematic controller combined with sliding mode control, J. Brazilian Soc. Mech. Sci. Eng. 46 (4) (2024) 238.
[29] R. Wang, X. Zou, J. Huang, X. Tian, A collision avoidance method for intelligent ship based on the improved bacterial foraging optimization algorithm, J. Robot. 2021 (1) (2021) 6661966.
[30] R. Ni, W. Zhu, C. Tao, Y. Cai, C. Qi, A Q-learning based multi-strategy integrated artificial bee colony algorithm with application in unmanned vehicle path planning, Expert Syst. Appl. 236 (2024) 121303.
[31] R. Yu, W. Luo, Reinforcement learning-based multi-strategy cuckoo search algorithm for 3D UAV path planning, Expert Syst. Appl. 223 (2023) 119910.
[32] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, W. Deng, Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning, Eng. Appl. Artif. Intell. 121 (2023) 105942.
[33] M.H. Nadimi-Shahraki, H. Zamani, Z.A. Varzaneh, A.S. Sadiq, S. Mirjalili, A systematic review of applying grey wolf optimizer, its variants, and its developments in different Internet of Things applications, Int. Things (2024) 101135.
[34] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82, https://doi.org/10.1109/4235.585893.
[35] F. Xu, et al., A new global best guided artificial bee colony algorithm with application in robot path planning, Appl. Soft Comput. 88 (2020) 106037.
[36] A. Mohammadzadeh, H. Taghavifar, Y. Zhang, W. Zhang, A fast nonsingleton type-3 fuzzy predictive controller for nonholonomic robots under sensor and actuator faults and measurement errors, IEEE Trans. Syst. Man, Cybern. Syst. (2024).
[37] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf optimizer, Adv. Eng. Softw. (2014), https://doi.org/10.1016/j.advengsoft.2013.12.007.
[38] M. Khishe, M.R. Mosavi, A. Moridi, Chaotic fractal walk trainer for sonar data set classification using multi-layer perceptron neural network and its hardware implementation, Appl. Acoust. 137 (2018), https://doi.org/10.1016/j.apacoust.2018.03.012.
[39] M.R. Mosavi, M. Khishe, M. Akbarisani, Neural network trained by biogeography-based optimizer with chaos for sonar data set classification, Wirel. Pers. Commun. 95 (4) (2017), https://doi.org/10.1007/s11277-017-4110-x.
[40] S. Ramasamy, K. Srikanth, S. Perumal, B. Perumal, F. Danielsen, Optimized mobile path planning algorithms considering energy, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2021, pp. 1–8.
[41] S. Kumar, A. Sikander, Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming," Arab. J. Sci. Eng. 47 (3) (2022) 3519–3539.
[42] M.A. Contreras-Cruz, V. Ayala-Ramirez, U.H. Hernandez-Belmonte, Mobile robot path planning using artificial bee colony and evolutionary programming, Appl. Soft Comput. 30 (2015) 319–328.
[43] K. Krishnamoorthy, "Wilcoxon signed-rank test," in *Handbook of Statistical Distributions With Applications*, 2020, pp. 339–342. doi: 10.1201/9781420011371-34.
[44] X. Chen, S. Zhang, Y. Liu, Q. Gao, Q. Gao, Manipulation of the flow induced by afterbody vortices using sweeping jets, Phys. Fluids 36 (4) (2024).
[45] B. Wang, L. Liu, Y. Li, M. Khishe, Robust Grey Wolf Optimizer for multimodal optimizations: a cross-dimensional coordination approach, J. Sci. Comput. 92 (3) (2022) 110.
[46] M. Taghavi and M. Khishe, "A modified grey wolf optimizer by individual best memory and penalty factor for sonar and radar dataset classification," 2019.
[47] M. Majidzadeh, A. Mahmoudzadeh, M. Khishe, Active sonar image classification using deep convolutional neural network evolved by robust comprehensive Grey Wolf optimizer, Neural Process. Lett. (2023) 1–24.
[48] D. Sun, D. Wu, A.E. Yamashita, F. Li, Fuzzified policy optimization with temporal velocity obstacle based collision avoidance path planning for multi unmanned surface vehicles, Ocean Eng 273 (2023) 114003.
[49] C.Y. Zhu, Intelligent robot path planning and navigation based on reinforcement learning and adaptive control, J. Logist. Inform. Serv. Sci. 10 (3) (2023) 235–248.
[50] T. Hang, et al., The optimal global path planning of mobile robot based on improved hybrid adaptive genetic algorithm in different tasks and complex road environments, IEEE Access (2024).