Enhancing Whale Optimization Algorithm with Differential Evolution and Lévy Flight for Robot Path Planning

Rongrong TANG*, Xuebang TANG, Hongwang ZHAO
Guilin University of Aerospace Technology, Guilin, Guangxi 541004, China

***Abstract*—Path planning is a prominent and essential part of mobile robot navigation in robotics. It allows robots to determine the optimal path from a given beginning point to a desired end goal. Additionally, it enables robots to navigate around obstacles, recognize secure pathways, and select the optimal route to follow, considering multiple aspects. The Whale Optimization Algorithm (WOA) is a frequently adopted approach to planning mobile robot paths. However, conventional WOA suffers from drawbacks such as a sluggish convergence rate, inefficiency, and local optimization traps. This study presents a novel methodology integrating WOA with Lévy flight and Differential Evolution (DE) to plan robot paths. As WOA evolves, the Levy flight promotes worldwide search capabilities. On the other hand, DE enhances WOA's ability to perform local searches and exploitation while also maintaining a variety of solutions to avoid getting stuck in local optima. The simulation results demonstrate that the proposed approach offers greater planning efficiency and enhanced route quality.**

Keywords*—*Path planning; mobile robot; differential evolution; Whale Optimization Algorithm; lévy flight

I. Introduction

Recent technological advancements, such as Artificial Intelligence (AI) [1], Machine Learning (ML) [2], and advanced sensor technologies [3], have significantly expanded the capabilities and applications of mobile robots. Initially, mobile robots were restricted to manufacturing industries. Still, reconsidering this concept led to their applicability in diverse fields, such as entertainment, health, mining, education, military, and agriculture [4]. Navigating the mobile robots is the most important phase, which can be defined as finding the robot's position, the best path for traveling. Localization is the first critical phase in navigation, wherein the robot should understand its position on the map of the real world. The path planning phase is the second key phase in which the robot calculates the route on the map of the surrounding environment [5]. Using this path, the robot reaches the goal and follows a strategic path. As a result, a well-designed map is essential to a successful navigation system, as it will enable the robot to reach its goal with the least amount of energy and time [6].

During the navigation task, robots use various cognitive devices to interpret their surroundings, orient themselves, regulate their actions, recognize obstacles, and avoid collisions using navigation strategies [7]. By acknowledging and sidestepping obstacles, navigation systems help an agent produce an accurate path from the start to the goal [8]. The selection of appropriate navigation technology for path planning is critical for robotic systems in simple and complex environments. Mobile robot navigation has been extensively studied in the past decade [9]. The navigation of mobile robots falls into three categories: personal, local, and global [10]. Global navigation is locating objects relative to a reference axis and progressing towards a specific goal [11]. Local navigation entails recognizing the changing conditions of the environment while identifying the spatial connections among various objects [12]. Personal navigation necessitates coordinating and adjusting several environmental factors that affect each other based on their respective positions [13]. Fig. 1 illustrates the fundamental operations of the robot.

The problem of path planning is classified as NP-hard due to its complex structure [14]. Heuristic and evolutionary algorithms are commonly employed to discover the best solution to this issue, particularly in extensive and complicated settings. One primary constraint in previous research is that many studies represent the context with discrete grids to determine the most effective grid configuration for determining the optimum path [15]. The primary limitation of this approach is the predetermined grid positions, which restrict path design flexibility. Furthermore, the A* algorithm can be used to identify optimum paths within arbitrary grids. Both the Dijkstra and A* algorithms are highly efficient because of their deterministic properties, which distinguishes them from evolutionary algorithms since they are not affected by the initial conditions. They exhibit significant time efficiency, especially compared to various evolutionary algorithms in two-dimensional path planning [16].

AI, ML, and Neural Networks (NNs) play pivotal roles in revolutionizing robot path planning. AI algorithms enable robots to navigate complex environments autonomously by leveraging advanced decision-making processes [17, 18]. ML techniques, particularly reinforcement learning, empower robots to learn from their experiences and optimize path planning strategies over time [19-21]. NNs, inspired by the human brain's structure, excel at pattern recognition and can efficiently process vast amounts of sensor data to make real-time navigation decisions [22-24]. Together, these technologies enhance the adaptability, efficiency, and reliability of robot path planning systems. The integration of these cutting-edge technologies not only addresses the challenges of traditional path planning methods but also paves the way for the next generation of intelligent robotic systems capable of seamlessly navigating diverse and challenging terrains [25, 26].

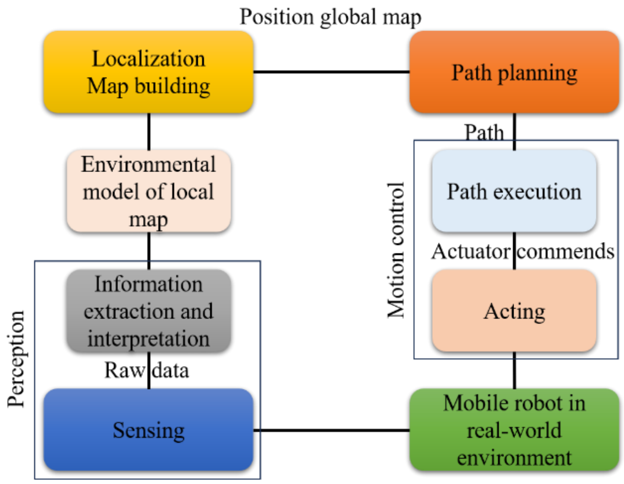


Fig. 1. Fundamental operations of mobile robot navigation.

The use and improvement of different meta-heuristic and hybrid algorithms for robot path planning is an emerging topic. The Whale Optimization Algorithm (WOA) is a well-known algorithm derived from whale hunting patterns [27]. The WOA uses two search methods to improve exploration and exploitability: shrinking the surrounding area and spiral updating the location to refine its search. WOA shows great promise by surpassing the performance of other established optimization techniques. Additionally, WOA features self-adjusting capabilities for some parameters as it goes through its iterations.

Like other metaheuristic algorithms, WOA has weaknesses, including premature convergence and vulnerability to becoming stuck in local optima. Overcoming these constraints constitutes a typical challenge to the advancements of metaheuristic searching. In the previous literature, many attempts have been under consideration to deal with these constraints, including using mathematical distributions using new evolutionary processes or combining different swarm intelligence techniques. This paper proposes an optimized iteration method of WOA for robot path planning. The enhanced method, named WOA-DELF, combines the techniques of differential evolution and Lévy flight. In WOA-DELF, Lévy flight is employed in the method exploring process to improve the optimization capability of global optimization. The system employs two distinct foraging approaches to optimize local conditions and incorporates Differential Evolution (DE) to enhance exploration of both local and global search areas during the exploitation phase.

The Lévy flight refers to a random walk distinguished by probability distributions with a high tail. It has been extensively utilized in several areas, including analyzing flying patterns in insects, feeding patterns in animals, and predicting human travel dispersion. The Lévy flight has been included in swarm intelligence approaches to search and optimize solutions. The addition of the Lévy flight improved both the exploration and exploitation of solutions [28]. Storn and Price developed Differential Evolution (DE) in 1995 to solve real-number optimization issues [29]. Over time, it has transformed into a versatile global optimization method deeply rooted in population dynamics. DE has gained recognition for its efficiency, success, robustness, and global search capabilities. Nevertheless, it shows limited local search capability and slightly sluggish convergence [30]. To tackle these difficulties, recent research has concentrated on promoting variety among populations, expanding exploration and exploitation capacities through parameter management, and preventing early convergence.

The remaining part of the paper is organized as follows. Section II reviews the related work on the problem of path planning for mobile robots. Section III defines the problem statement. Section IV introduces the proposed approach, outlining the way WOA is used in combination with DE and LF for improving path planning. Section V presents the results of simulations conducted to confirm the efficiency of the presented method. Section VI draws the paper to a close and suggests areas for further research.

II. Related Work

This section aims to offer a comprehensive review of the state of the art in mobile robot route planning by comparing noteworthy studies compiled from relevant literature. The purpose is to describe the diverse techniques applied to mobile robot navigation and route optimization to address its associated challenges. Table I presents the essential characteristics of each study, such as the employed methodology, fundamental procedures, main goals, assessment measures, and significant discoveries.

Ajeil, et al. [31] tackled the problem of path planning for self-moving mobile robots in stable and varying settings. Their objective was to find a trajectory that is devoid of collisions and fulfills the criteria of being the shortest distance and smooth. The proposed method effectively simulates a real-world scenario by taking into account the physical attributes of mobile robots. The problem is presented as the motion of a point in an empty space. There are three components to the algorithm. The first part creates an efficient route by utilizing a hybridized PSO-MFB method, which incorporates Modified Frequency Bat (MFB) and PSO algorithms to reduce path length and ensure smooth navigation. The second part identifies all incorrect values produced by the hybrid algorithm and employs a unique local search method to transform points into valid results. The third component is equipped with obstacle sensors and collision avoidance, triggering as the robot detects obstructions within its sensor range, keeping it from colliding. The numerical results indicate that the proposed method generates an optimal and feasible path in complex and dynamic scenarios, surpassing the limitations of traditional grid-based approaches.

Das and Jena [33] presented a novel method for calculating the best collision-free paths for individual robots in simple and intricate surroundings. They resolved the problem by using an improved version of the PSO algorithm combined with evolutionary operators (EOPs). The improvement of the PSO algorithm included incorporating the concept of governance in human society and two evolutionary multi-crossover operators from the genetic algorithm and the bee colony operator to boost the intensification capability of the PSO algorithm. The technique was created to calculate the deadlock-free sequence coordinates of each robot using their current coordinates. The goal was to decrease the distance traveled by each robot by finding a middle ground between intensity and diversity. The study results verified that PSO-EOPs exhibited superior performance compared to PSO and DE in terms of efficiency in arrival time, secure route formation, and energy consumption during travel.

TABLE I. OVERVIEW OF ROBOT PATH PLANNING APPROACHES

Reference	Approach	Primary techniques	Main objectives
[31]	Path planning for mobile robots	Hybridized PSO-MFB method and obstacle sensors	Collision-free, lowest distance, and smooth paths
[32]	Obstacle avoidance for multi-robot systems	Covariance matrix adaptation evolution and proximal policy optimization	Avoid obstacles
[33]	Collision-free paths for individual robots	Improved PSO algorithm and evolutionary operators	Arrival time, secure route construction, and energy usage
[34]	Mobile robot route planning	Multi-goal Genetic Algorithm (MOGA)	Safety, distance, smoothness, traveling time, and collision-free path
[35]	Mayfly optimization algorithm for robot route planning	Improved mayfly optimization algorithm based on q-learning	Global search capabilities and avoidance of local optima
[36]	Path planning for multiple robots	Enhanced artificial bee colony algorithm and ABCL method	Optimal collision-free courses for multiple robots

Suresh, et al. [34] proposed the Mobile Robot route Search powered by a Multi-goal Genetic Algorithm (MRPS-MOGA), a new method that uses a genetic algorithm with various goal functions to solve mobile robot route planning issues. The primary purpose of MRPS-MOGA is to determine the most efficient route by taking into account five specific criteria: safety, distance, smoothness, trip time, and avoidance of collisions. The multi-objective genetic algorithm (MOGA) is used to choose the best route among several possible options. The population is created with randomly generated routes, and fitness values are assessed using different objective functions. The fitness criteria decide whether routes are kept for involvement in the following generation. The MRPS-MOGA approach utilizes genetic algorithm components such as tournament selection, ring crossover, and adaptive bit string mutation to find the best path. A mutation operator is randomly applied to the sequence to introduce variation in the population. An evaluation of the individual fitness criteria is conducted to ascertain the optimal course of action for the population. The MRPS-MOGA algorithm was evaluated in multiple scenarios, proving its superiority in choosing the most efficient route while minimizing time complexity. The experimental research has shown that MRPS-MOGA is a highly effective method for designing paths for mobile robots. It offers enhanced safety, reduced energy usage, and faster transit times in comparison to existing techniques.

Zou, et al. [35] have discussed issues in the fundamental Mayfly Optimization Algorithm (MOA) for robot route planning, such as sluggish convergence, low precision, instability, and applicability limited to static situations. A fusion technique was suggested that merges an enhanced Mayfly Optimization technique with the Dynamic Window Approach. An Improved Mayfly Optimization Algorithm based on Q-learning (IMOA-QL) is presented for global robot path planning. The new algorithm's primary function is Q-learning, which adjusts parameters dynamically to boost global search capabilities and prevent becoming stuck in local optima. Global path nodes are recovered as sub-target points, and the Dynamic Window Approach is used to plan local paths to increase real-time avoidance capabilities. IMOA-QL's efficacy is confirmed by 20 random simulation trials in a 100×100 static map scenario, where it is compared with basic MOA and MOA-LAIW. IMOA-QL decreases the average path length by 4.4% and 2.1% compared to MOA and MOA-LAIW in simple settings and by 6.5% and 3.2% in complex environments, as shown by the results. In 20 studies, the average variance of IMOA-QL decreased by 74.1% and 57.6% in simple contexts and by 51.2% and 38.6% in complex environments compared to MOA and MOA-LAIW.

Wen, et al. [32] developed a flexible optimization method based on covariance matrix adaptation evolution, derived from the traditional proximal policy optimization, to develop an effective obstacle avoidance strategy for autonomous navigation of multi-robot systems in complicated situations with static and dynamic obstacles. The test outcomes indicated that the proposed method was effective for avoiding obstacles and achieving the goal location. Meta-learning was combined with multi-robot architectures to enhance their flexibility. The proposed method was utilized in the training of robots to acquire a multi-task policy.

Cui, et al. [36] explored path planning for multiple robots in an ongoing familiar environment, introducing an innovative method for local path planning. They created a new way to implement metaheuristic algorithms to design optimal collision-free courses for multiple robots and enhance the Artificial Bee Colony (ABC) algorithm. Three enhancements have been included in the ABC algorithm in this scenario. The search equations of the deployed bee and scout bee phases were improved by including the global best individual to improve control over the search direction. The learning mechanism of the TLBO algorithm was introduced into the spectator bee phase to enhance exploitability. The ABCL method, based on learning, was utilized to calculate the next locations of all robots by considering their present coordinates, path length, safety, and planning efficiency. ABCL outperformed seven effective metaheuristic algorithms in tackling diverse optimization problems, as demonstrated in experimental investigations on benchmark functions. Simulation experiments for multi-robot route planning demonstrated that ABCL surpasses its competitors in producing optimal collision-free pathways and runtime. ABCL enhanced two features by an average of 2.1% and 12.6% compared to the original ABC across all tasks. Thus, the suggested implementation technique demonstrates that

ABCL is an efficient option for path planning for numerous robots.

III. Problem Statement

The current landscape of robotics demands efficient path planning in large environments, taking into account computational limitations. Memory constraints may render finding an optimal path impractical, particularly when dealing with expansive navigation spaces. This challenge intensifies when multiple criteria, such as path length, distance to obstacles, and search complexity, must be considered for global path efficiency in cluttered environments. Different regions of large, cluttered maps may elicit varying responses from fixed path-planning algorithms, making it difficult to achieve universal efficiency across all conditions.

Path planning is the process of determining a limited number of possible motions within an unobstructed area of a design, usually from predetermined starting to end points. While multiple paths may exist, path-planning algorithms aim to find the optimal path based on predefined objective functions, such as minimizing path length, maximizing smoothness, or ensuring safety.

This study introduces a novel path-planning method aimed at identifying the most efficient routes in various intricate settings between specified source and target points. The method assesses path quality by considering factors such as route length, smoothness, and safety. The study examines 2D settings with stationary barriers of various shapes, assuming no relationship between obstacles and free space. Robots are considered single entities, taking into account their dimensions by including a confidence radius near objects. Multi-robot path planning scenarios assume that each robot moves at a constant speed.

The technique creates a map of the environment, making it easier to find possible segmented linear pathways between the starting and target locations in a gridded area. It ensures the identification of at least one viable route if it is present. Subsequently, the algorithm identifies the appropriate positions of Path Bases (PBs) selected grids used to determine the paths. These PBs can then be connected using cubic spline or piecewise linear methods to construct optimal paths. Fig. 2 shows an outline of the suggested technique for optimal path planning in an ongoing area.

Path planning in ongoing areas with variable impediments might be computationally difficult due to many issues. Researchers simplify the challenge by transforming it into identifying a finite sequence of hops in a gridded context between origin and target points. However, these approaches are restricted by the degree of separation.

The algorithm developed in this study uses multiple methodologies to model the situation. The surrounding area is divided into grid-like squares. Fig. 3 demonstrates a 1010 field separated into squares of one unit length. 100 points, represented as green circles, are evenly spread over the region. The robot's trajectory is determined by choosing a suitable group of nodes inside the gridded setting. The method assigns potential values to all points, and possible pathways connecting the source and target points are found using pre-calculated potentials.

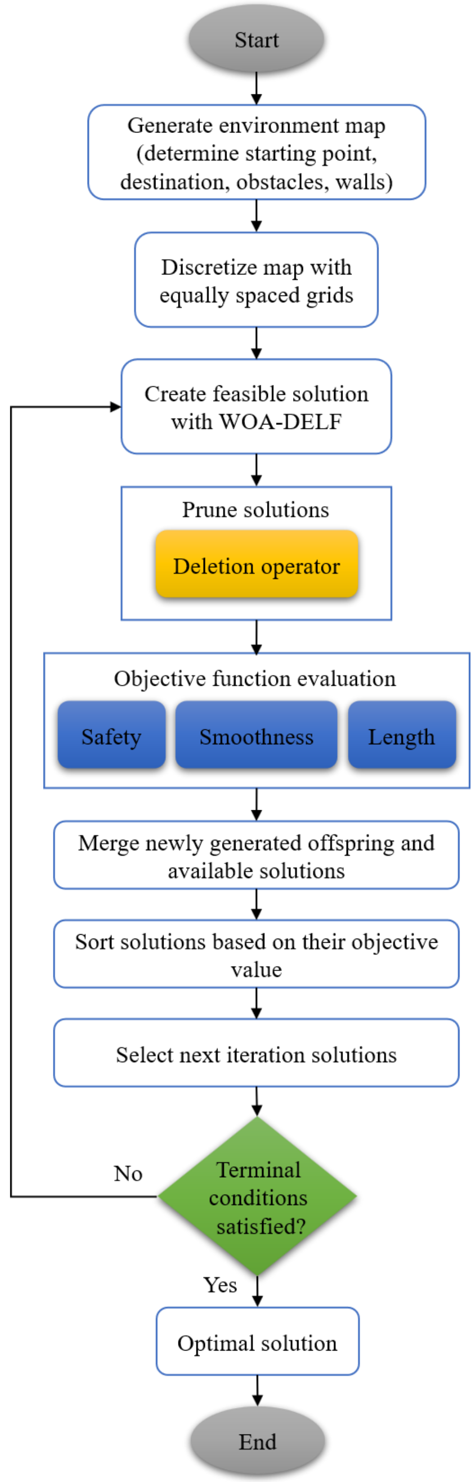


Fig. 2. Suggested technique for optimal path planning in continuous environments.

Fig. 3 depicts the chosen PBs connected by a blue dashed line, representing the potential route from the beginning point to the destination. The coordinates of the potential starting points associated with each conceivable initial route are encoded as solutions. Fig. 4 illustrates the solution layout related to the beginning route seen in Fig. 3.

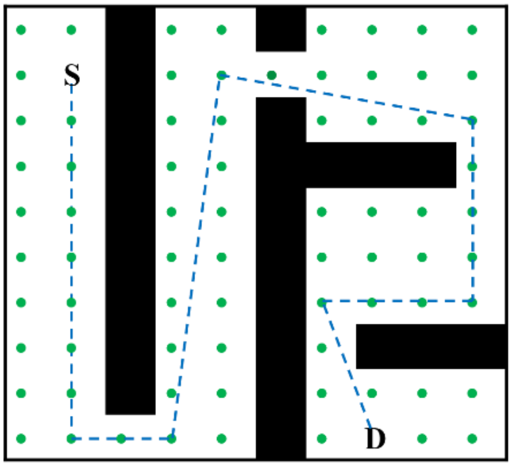


Fig. 3. Path planning process with potential bases.

1	3	5	9	10	7
1	2	9	9	7	4

Fig. 4. Solution layout for identified route.

The suggested approach assigns the node representing the target location the highest potential. The potential lowers steadily as one travels to nearby nodes. The neighboring points for each point reside at a distance d√2 from the point, where d refers to the number of discretizations. This technique creates a diagram displaying the possibilities of the area. This potential map may identify all possible routes between the starting point and the goal. Fig. 5 shows the potential map for the setting shown in Fig. 3. The potential map of the proposed approach is constructed using three lists of nodes: CLOSED, TEMP, and OPEN.

- CLOSED list: It is composed of potential nodes and their adjacent nodes.
- TEMP list: This list includes points given a potential, with the condition that their neighboring points are not assigned potentials.
- OPEN list: Points that have not been assigned a potential are included in this list.

The algorithm initializes by inserting all nodes into the OPEN list. The process then involves the following steps:

- Step 1: The target point is eliminated from the OPEN list, assigned the highest potential (e0), and inserted into the TEMP list.
- Step 2: Obstacle points are eliminated from the OPEN list, assigned a potential of -e0, and added to the CLOSED list.
- Step 3: Assuming the point of departure, nodes adjacent to it are assigned a potential of e1 (e1 = e0 - a, where 'a' is the decrement step), inserted into the TEMP list, and excluded from the OPEN list. The target node is deleted from the TEMP list and added to the CLOSED list.
- Step 4: In each subsequent round (ith iteration), points in the TEMP list have their neighboring points given potential values of ei (ei = ei-1 - a), are moved to the CLOSED list, and points accepting potential values move from the OPEN list to the TEMP list.

The repetition of these steps results in a possible representation of the area. Feasible initial paths are then determined by selecting adjacent nodes with the highest potential starting from the start location. This process gradually increases the route's potential until it reaches the final point, which has the largest potential. The algorithm guarantees the finding of possible initial paths, and in particular instances, paths may be divided into sub-paths when two nearby points of a single point are equal in potential. Fig. 5 illustrates the potential map of the environment, and Fig. 6 illustrates three possible routes resulting from the suggested algorithm.

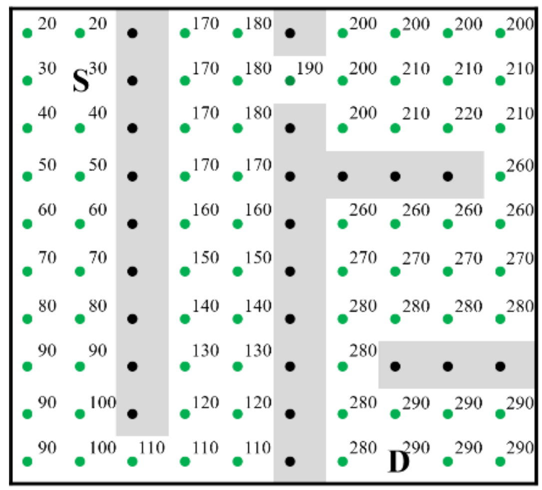


Fig. 5. Potential map construction.

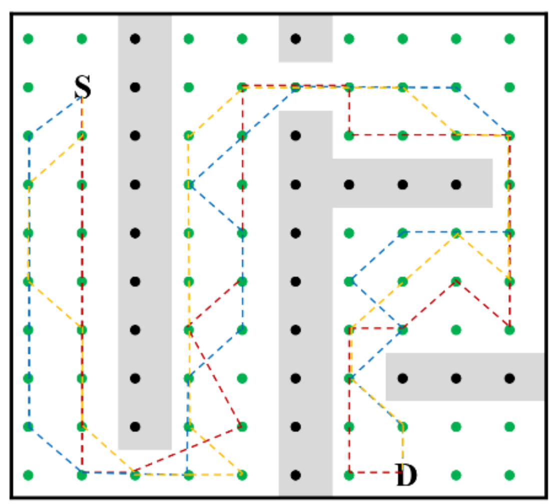


Fig. 6. Possible routes generated by the algorithm.

IV. PROPOSED TECHNIQUE

The study combines WOA with Lévy flight and DE to optimize PB position within a continuously changing context. The algorithm's evolution is an iterative process, and the optimization continues until the algorithm reaches its final state. In this context, the final state is regarded as arriving at a fixed number of iterations. After the specified number of iterations is finished, the optimization process terminates and the final solution is found. The predetermined number of iterations acts as a termination condition for the algorithm's execution.

The WOA draws inspiration from the foraging behavior of whales, particularly the hunting strategies observed in humpback whales. This algorithm emulates three distinct foraging behaviors, mirroring the actions of humpback whales: encircling prey, bubble net assaulting, and randomly hunting prey. These behaviors are represented by mathematical models in order to accurately reflect the fundamental aspects of whale hunting strategies. Humpback whales have the capacity to detect prey that is close by and position themselves strategically in the group to take advantage of the recognized prey location, which is regarded the most advantageous position. While closing in on the prey, continuous adjustments are made to their positions. In the WOA context, the algorithm perceives the resulting viable solutions as 'whales' and designates the present most optimal solution or local optimum for encircling prey. The algorithm employs a function to represent prey encirclement, as expressed in Eq. (1).

$$\vec{X}(t+1) = \vec{X}_{best}(t) - \vec{A} \cdot |\vec{C} \cdot \vec{X}_{best}(t) - \vec{X}(t)| \quad (1)$$

In Eq. (1), X signifies the chosen search whale, $\cdot$ denotes element-wise multiplication, $\vec{X}_{best}(t)$ refers to the best position of a whale in the current iteration t, and $|\vec{C}.\vec{X}_{best}(t) - \vec{X}(t)|$ indicates the distance between $\vec{C}.\vec{X}_{best}(t)$ and $\vec{X}(t)$. The coefficient vectors $\vec{A}$ and $\vec{C}$ have varying characteristics, and their updates are governed by Eq. (2) and Eq. (3), respectively.

$$\vec{A} = 2 \times \vec{a} \times \vec{r} - \vec{a} \quad (2)$$

$$\vec{C} = 2 \times \vec{r} \quad (3)$$

The vector $\vec{a}$ gradually decreases from 2 to 0 following the formula max $2 - 2t/t_{max}$, where t_{max} is the maximum iteration count. $\vec{r}$ represents a stochastic vector with values between 0 and 1, restricting the values of $\vec{A}$ to fall within the range of $[-\vec{a}, \vec{a}]$. It is crucial to note that the random vectors $\vec{A}$ and $\vec{C}$ are essential in directing the whale to adjust its location in order to reach the ideal solution. Humpback whales utilize bubble nets to herd and trap animals near the water's surface as part of their normal habit. The mathematical model of the spiral bubble net assault method is expressed by Eq. (4).

$$\vec{X}(t+1) = |\vec{X}_{best}(t) - \vec{X}(t)|.e^{bl}\cos(2\pi l) + \vec{X}_{best}(t) \quad (4)$$

The parameter b defines the logarithmic spiral form, with l being a randomly chosen value between 0 and 1. In the natural behavior of humpback whales, exploration of new target prey involves randomly selecting a whale position and swimming towards it. The formula employed in the WOA is designed to simulate this process for global search.

$$\vec{X}(t+1) = \vec{X}_{rand}(t) - \vec{A} \cdot |\vec{C} \cdot \vec{X}_{rand}(t) - \vec{X}(t)| \quad (5)$$

Operator selection is controlled by a random switch control parameter, p, ranging from [0, 1]. The vector $\vec{A}$ plays a crucial role in determining the hunting method of the whale. If we assume a 50% probability for the whale to choose the bubble-net attacking method during the position update for solution exploitation, the likelihood of selecting the operator when hunting or encircling prey is additionally influenced by the adaptive variation of the vector $\vec{A}$. Eq. (6) expresses a formula for selecting operators.

During the exploration stage of the WOA, individuals update their positions by sharing information with another individual in a limited solution space. The exploration phase of WOA incorporates Lévy flight to improve global search capabilities and speed up convergence. Lévy flight involves sporadic huge steps or extended leaps, which widen the exploration area. The position of humpback whales is updated using the step of Lévy flight, as described by Eq. (7).

$$\vec{X}(t+1) = \begin{cases} \vec{X}_{best}(t) = \vec{A}.\left|\vec{C}.\vec{X}_{best}(t) - \vec{X}(t)\right|, if\ p < 0.5\ and |\vec{A}| < 1 \\ \vec{X}(t+1) = \vec{X}_{best}(t) - \vec{A}.\left|\vec{C}.\vec{X}_{rand}(t) - \vec{X}(t)\right|, if\ p < 0.5\ and\ |\vec{A}| \geq 1 \\ \vec{X}_{best}(t) = \vec{X}(t).e^{bl}\cos(2\pi l) + \vec{X}_{best}(t), if\ p \geq 0.5 \end{cases} \quad (6)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}}(t) + a.sign[rand - 1/2] \oplus Levy(s) \quad (7)$$

The $sign[rand - 1/2]$ term takes values of -1, 0, or 1, the $\oplus$ symbol represents entry-wise multiplication, and a denotes the step size determined by Eq. (8).

$$a = a_0[\overrightarrow{X_{rand}}(t) - \vec{X}(t)] \quad (8)$$

Here, a_0 is set to 0.01, and $\overrightarrow{X_{rand}}$ measures the position vector of a whale chosen at random. Lévy flight follows a Lévy distribution for the step length, given by:

$$Levy(s) \sim |s|^{-1-\beta}, \quad 0 < \beta \leq 2 \quad (9)$$

In this expression, β is set to 1.5, and μ and ν have a normal distribution. The complete calculation of s involves Mantega's algorithm.

$$s = \frac{\mu}{|\nu|^{1/\beta}}, \ \mu \sim N(0, \sigma_\mu^2), \ \nu \sim N(0, \sigma_\nu^2) \quad (10)$$

$$\sigma_\mu = \left\{\frac{\Gamma(1+\beta).\sin(\pi\beta/2)}{\beta.\Gamma\left[\frac{1+\beta}{2}\right].2^{(\beta-1)/2}}\right\}^{1/\beta}, \sigma_\nu = 1 \quad (11)$$

Finally, Eq. (6) can be rewritten as:

$$\vec{X}(t+1) = \overrightarrow{X_{rand}}(t) + sign\left[rand - \frac{1}{2}\right] . a_0 \frac{\mu}{|\nu|^{1/\beta}} . [\overrightarrow{X_{rand}}(t) - \vec{X}(t)] \quad (12)$$

In the final phase of the WOA, individual positions are updated through a greedy selection operation limited to the best solution. This limitation makes it vulnerable to getting trapped in local optima. To solve this problem, DE is incorporated into WOA. A set of external archives is created consisting of individual populations and historically optimal populations. In each iteration, new solutions are modified with DE search strategies in accordance with the external archive set. This integration improves the exchange of information between individual solutions and improves WOA's local search and exploitation capabilities.

DE involves an external archive set, NP D-dimensional individuals represented as $x_{i,G} = \{x_{i,G}^1, \ldots, x_{i,G}^D\}$, where $i = 1, \ldots, NP$, and G is the number of generations. Each dimension of the individual is constrained by $x_{min} = \{x_{min}^1, \ldots, x_{min}^D\}$ and $x_{max} = \{x_{max}^1, \ldots, x_{max}^D\}$. The initial population is usually randomly generated in the feasible region.

The mutation operator generates mutant vectors $\nu_{i,G}$, where DE/rand/1 is a commonly used operator. The generated $\nu_{i,G}$ can be expressed as

$$\nu_{i,G} = x_{r1,G} + F * (x_{r2,G} - X_{r3,G}), r1 \neq r2 \neq r3 \neq i \quad (13)$$

Here, $x_{r1,G}$, $x_{r2,G}$, and $x_{r3,G}$ are chosen from the current population, and F is the mutation control parameter that scales the difference vector. Different mutation strategies can be employed, with DE/rand/1 being one of the variants.

DE/rand/1: $\nu_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G})$

DE/best/1: $\nu_{i,G} = X_{best,G} + F(X_{r2,G} - X_{r3,G})$

DE/current/1: $\nu_{i,G} = X_{i,G} + F(X_{r2,G} - X_{r3,G})$

DE/current-to-best/1: $\nu_{i,G} = X_{i,G} + F(X_{best,G} - X_{i,G}) + F(X_{r1,G} - X_{r2,G})$

DE/rand/2: $\nu_{i,G} = X_{r1,G} + F(X_{r2,G} - X_{r3,G}) + F(X_{r4,G} - X_{r5,G})$ (14)

DE/best/2: $\nu_{i,G} = X_{best,G} + F(X_{r1,G} - X_{r2,G}) + F(X_{r3,G} - X_{r4,G})$

DE/current-to-rand/1: $\nu_{i,G} = X_{i,G} + F(X_{r1,G} - X_{i,G}) + F(X_{r2,G} - X_{r3,G})$

DE/current-to-pbest/1: $\nu_{i,G} = X_{i,G} + F(X_{best,G}^p - X_{i,G}) + F(X_{r1,G} - \tilde{X}_{r2,G})$

$X_{best,G}^p$ refers to the individual with the optimal fitness function value at the G^{th} generation. The binomial crossover operator, commonly employed, can be chosen to generate the trail vector $u_{i,G}$ between $x_{i,G}$ and $\nu_{i,G}$, as expressed by the formula below (see Eq. (15)). $rand_j$ is a randomly generated number uniformly distributed within the range [0, 1]. $CR_i \in (0, 1)$ serves as the crossover control parameter, and n_j is a randomly generated integer within the range $[1, D]$.

$$u_{i,G}^j = \begin{cases} \nu_{i,G}^j, \ \ if\ rand_j \leq CR_i\ or\ j = n_j \\ \ i = 1, 2, \ldots, NP; \ \ j = 1, 2, \ldots, D \\ \ x_{i,G}^j \ \ otherwise \end{cases} \quad (15)$$

Subsequently, a superior individual between the trail vector $u_{i,G}$ and target vector $x_{i,G}$ will be chosen. The superior individual will persist into the next generation based on a comparison of the fitness values, employing greedy selection as outlined in Eq. (16). The fitness function values of the target vector $x_{i,G}$ and trail vector $u_{i,G}$ are denoted by $f(x_{i,G})$ and $f(u_{i,G})$, respectively.

$$X_{i,G+1} = \begin{cases} u_{i,G+1}, \ if\ f(u_{i,G}) \leq f(X_{i,G}) \\ X_{i,G}, \ \ otherwise \end{cases} \quad (16)$$

The "DE/rand/1" method tends to enhance exploration but with sluggish convergence speeds. The "DE/best/1" method typically exhibits rapid convergence but lacks exploitation capabilities and is prone to getting trapped near local optimum. "DE/current-to-rand/1" offers more diverse populations and global search capabilities, but comes with certain drawbacks like perturbation and blindness. Conversely, "DE/current-to-pbest/1" excels in search stability and exploitation ability.

In the DE algorithm, we opt for the "DE/current-to-pbest/1", an archive-based hybrid memory evolutionary operator. Commencing the search for the existing individual and employing multiple local optimizations to guide it results in better individual diversity, avoiding premature convergence to local optima. WOA deep exploitation becomes more stable as a result. This study fine-tunes variables F and CR, incorporating "DE/current-to-pbest/1" to aid WOA in navigating local areas, capturing prey, and improving overall stability. The WOA-DELF integrates DE and Lévy flight into the fundamental WOA.

V. Experimental Results

The test function is an important indicator in measuring the performance of the algorithm to find the best solution. The smaller the value is under the same conditions, the better the searching and developing ability. There are eight test functions used in the experiment to verify the efficiency of the algorithm. The functions can be categorized into two distinct groups: unimodal and multi-modal, as illustrated in Table II.

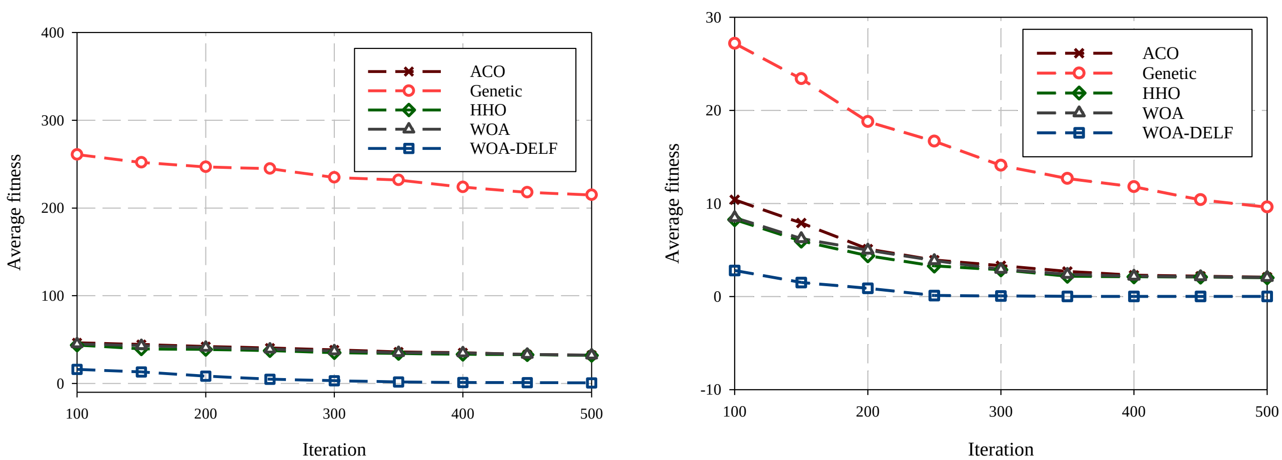
The unimodal functions (f1-f5) are used to investigate algorithm exploitation capabilities, since they have only a single global minimum without any local minimum. On the other hand, multi-modal functions (f6-f8) are employed to analyze the ability of the algorithm to search for different local minima and to avoid all local minima.

TABLE II. NUMERICAL FUNCTIONS

Functions	Ranges	F_{min}
$f_1(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30,30]	0
$f_2(x) = \max\{\lvert x_i \rvert, 1 \le i \le n\}$	[-100,100]	0
$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	[-100,100]	0
$f_4(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	[-10,10]	0
$f_5(x) = \sum_{i=1}^{n} x_i^2$	[-100,100]	0
$f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + c$	[-32,32]	0
$f_7(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	[-5.12,5.12]	0
$f_8(x) = \sum_{i=1}^{n} x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)$	[-500,500]	0

In the comparison, WOA-DELF is evaluated alongside four other algorithm types: WOA, ACO, genetic, and HHO algorithms. WOA maximizes the efficacy of a robot's trajectory by mimicking the social behavior of humpback whales. As it keeps on improving the candidate solutions, WOA systematically explores the solution space. Ultimately, it improves the ease of robot mobility through specific types of terrains and increases path length. Inspired by the foraging behavior of ants, the ACO algorithm is well-suited for robot path planning. Artificial ants represent potential paths, depositing pheromones on explored routes. With accumulated time, the paths with more pheromones guide the robot to the efficient navigation of environment. In the field of robotics, Genetic Algorithms are used for robot path planning. Initially a set of potential paths is generated and refined step by step through crossover and mutation, thereby modelling the survival of the fittest. As the algorithm iterates, the robot adapts its trajectory to the immediate surroundings. HHO, inspired by harmony in music, is employed to improve the path of the robot by adjusting the variables within the solution space. The harmony seeking algorithm aims to maintain a harmony between exploration and exploitation enabling a robot to find out the paths efficiently without any obstacles.

This experiment ran 500 iterations with 30 search agents for 8 test functions, each with 30 dimensions. As depicted in Table III, for the unimodal test functions (F1-F5), the WOA-DELF algorithm performs better than other algorithms, namely WOA and DELF, which proves that the search space is well utilized. In multi-modal functions (F6-F8), the functions which are difficult to obtain because of multiple and local optima presence, the WOA-DELF algorithm consistently outperforms other algorithms. As shown in Table III and Fig. 7, the proposed algorithm has significant advantages over all other alternatives. The results are the same when performing standard deviation tests.



(a) (b)

Fig. 7. Curve chart based on test functions: (a) f_1 and (b) f_2.

TABLE III. PERFORMANCE COMPARISON OF ALGORITHMS ON TEST FUNCTIONS

Function	Fitness	ACO	Genetic	HHO	WOA	WOA-DELF
f_1	Max	$8.31\times10^{+01}$	$2.05\times10^{+01}$	$7.95\times10^{+01}$	$8.19\times10^{+01}$	24.81
	Min	$1.17\times10^{+01}$	$1.28\times10^{+03}$	$1.16\times10^{+01}$	$1.18\times10^{+01}$	5.12×10^{-16}
	Average	$3.22\times10^{+01}$	$2.15\times10^{+02}$	$3.18\times10^{+01}$	$3.21\times10^{+01}$	0.51
	Std	$1.94\times10^{+01}$	$2.92\times10^{+02}$	$1.92\times10^{+01}$	$1.95\times10^{+01}$	2.58
f_2	Max	$4.96\times10^{+00}$	$1.82\times10^{+01}$	$4.91\times10^{+00}$	$4.94\times10^{+00}$	8.21×10^{-245}

	Min	8.79×10^{-01}	$3.53\times10^{+00}$	8.71×10^{-01}	8.75×10^{-01}	0
	Average	$2.07\times10^{+00}$	$9.62\times10^{+00}$	$2.01\times10^{+00}$	$2.04\times10^{+00}$	8.24×10^{-247}
	Std	8.77×10^{-01}	$3.24\times10^{+00}$	8.71×10^{-01}	8.76×10^{-01}	0
f_3	Max	$3.69\times10^{+02}$	$2.89\times10^{+03}$	$3.66\times10^{+02}$	$3.68\times10^{+02}$	0
	Min	$2.53\times10^{+01}$	$1.98\times10^{+02}$	$2.51\times10^{+01}$	$2.54\times10^{+01}$	0
	Average	$1.27\times10^{+02}$	$1.31\times10^{+03}$	$1.26\times10^{+02}$	$1.27\times10^{+02}$	0
	Std	$8.18\times10^{+01}$	$7.25\times10^{+02}$	$8.15\times10^{+01}$	$8.17\times10^{+01}$	0
f_4	Max	8.68×10^{-03}	$3.19\times10^{+00}$	$4.97\times10^{+01}$	8.54×10^{-05}	6.67×10^{-28}
	Min	9.71×10^{-05}	1.08×10^{-01}	$2.65\times10^{+01}$	2.92×10^{-29}	0
	Average	1.51×10^{-03}	$1.41\times10^{+00}$	$3.41\times10^{+01}$	3.77×10^{-06}	6.67×10^{-27}
	Std	1.98×10^{-03}	9.47×10^{-01}	$5.82\times10^{+00}$	4.76×10^{-06}	0
f_5	Max	4.11×10^{-05}	6.74×10^{-07}	$1.18\times10^{+04}$	9.86×10^{-51}	0
	Min	7.42×10^{-08}	1.62×10^{-07}	$3.27\times10^{+03}$	8.66×10^{-07}	0
	Average	2.31×10^{-06}	1.34×10^{-07}	$6.33\times10^{+03}$	8.72×10^{-11}	0
	Std	7.28×10^{-06}	1.58×10^{-07}	$1.81\times10^{+03}$	2.73×10^{-09}	0
f_6	Max	$2.24\times10^{+00}$	$3.54\times10^{+00}$	$1.18\times10^{+01}$	2.51×10^{-05}	$8.82\times10^{+16}$
	Min	3.71×10^{-05}	$1.33\times10^{+00}$	$8.32\times10^{+00}$	8.92×10^{-15}	$8.82\times10^{+16}$
	Average	$1.12\times10^{+00}$	$2.11\times10^{+00}$	$1.13\times10^{+01}$	7.41×10^{-08}	$8.82\times10^{+16}$
	Std	6.58×10^{-01}	5.12×10^{-01}	7.75×10^{-01}	1.12×10^{-06}	0
f_7	Max	$7.67\times10^{+01}$	$7.98\times10^{+01}$	$1.91\times10^{+02}$	3.42×10^{-06}	0
	Min	$2.39\times10^{+01}$	$2.33\times10^{+01}$	$1.27\times10^{+02}$	$0.00\times10^{+00}$	0
	Average	$3.95\times10^{+01}$	$4.78\times10^{+01}$	$1.61\times10^{+02}$	3.42×10^{-09}	0
	Std	$1.34\times10^{+01}$	$1.49\times10^{+01}$	$1.57\times10^{+01}$	1.02×10^{-07}	0
f_8	Max	$-3.44\times10^{+03}$	$-4.81\times10^{+03}$	$-1.82\times10^{+03}$	$-4.88\times10^{+03}$	$-7.13\times10^{+03}$
	Min	$-6.35\times10^{+03}$	$-7.19\times10^{+03}$	$-3.21\times10^{+03}$	$-1.26\times10^{+04}$	$-8.69\times10^{+03}$
	Average	$-5.31\times10^{+03}$	$-5.91\times10^{+03}$	$-2.19\times10^{+03}$	$-7.41\times10^{+03}$	$-8.43\times10^{+03}$
	Std	$6.38\times10^{+02}$	$6.21\times10^{+02}$	$3.21\times10^{+02}$	$9.21\times10^{+02}$	386.18

VI. Conclusion

In this paper, we proposed an enhanced WOA by incorporating differential evolution and Lévy flight for robot path planning. Traditional WOA has a slow convergence, a low efficiency, and is easily trapped into local optima. Our improved WOA simultaneously has the ability to overcome these problems in classical WOA, which can effectively enhance the performance of WOA in robot path planning. Lévy Flight is used in the hybridization of WOA to maximize exploration throughout the evolutionary process, while DE is responsible for the exploitation that allows the algorithm to explore complex environments without being trapped in a local optimum. The simulation outcomes, performed over several unimodal and multi-modal benchmark test functions, revealed the effectiveness of the WOA-DELF algorithm compared to competing benchmarks, like the WOA, ACO, genetic, and HHO algorithms. WOA-DELF was also able to exploit the search space effectively on unimodal functions, as it delivered better planning efficiency and better route quality. In addition, WOA-DELF outperformed the other algorithms equally well on multimodal functions. This aspect further suggests that the exploration of WOA-DELF is desirable. The proposed algorithm's success across all tested scenarios and its favorable comparison against existing algorithms confirm its potential as an effective tool for robot path planning. The enhanced performance of the proposed algorithm in the experiments requires it to further optimize, test scalability, and deploy on real world scenarios for confirming its effectiveness in practical robotic navigation.

Acknowledgments

This work was supported by the project of Yulin City Central Leading Local Science and Technology Development Special Fund Project (20223402), Key research and development Plan Projects in Guilin City (20220106-3), Young and Middle-Aged Teachers in Guangxi Universities (2021KY0792).

References

[1] A. Behfar and H. Asadollahi, "Calculating optimal number of nodes for last Corona in q-switch method," International Journal of Computer Science and Information Security, vol. 14, no. 12, p. 786, 2016.

[2] F. Kamalov, B. Pourghebleh, M. Gheisari, Y. Liu, and S. Moussa, "Internet of Medical Things Privacy and Security: Challenges, Solutions, and Future Trends from a New Perspective," Sustainability, vol. 15, no. 4, p. 3317, 2023, doi: https://doi.org/10.3390/su15043317.

[3] J. Zandi, A. N. Afooshteh, and M. Ghassemian, "Implementation and analysis of a novel low power and portable energy measurement tool for wireless sensor nodes," in Electrical Engineering (ICEE), Iranian Conference on, 2018: IEEE, pp. 1517-1522, doi: 10.1109/ICEE.2018.8472439.

[4] P. K. Panigrahi and S. K. Bisoy, "Localization strategies for autonomous mobile robots: A review," Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 8, pp. 6019-6039, 2022, doi: 10.1016/j.jksuci.2021.02.015.

[5] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," Computer communications, vol. 149, pp. 270-299, 2020, doi: 10.1016/j.comcom.2019.10.014.

[6] S. Shokouhi, B. Mu, and M.-W. Thein, "Optimized Path Planning and Control for Autonomous Surface Vehicles using B-Splines and Nonlinear Model Predictive Control," in OCEANS 2023-MTS/IEEE US Gulf Coast, 2023: IEEE, pp. 1-9, doi: 10.23919/OCEANS52994.2023.10337066.

[7] F. Jacob, E. H. Grosse, S. Morana, and C. J. König, "Picking with a robot colleague: A systematic literature review and evaluation of technology acceptance in human–robot collaborative warehouses," Computers & Industrial Engineering, p. 109262, 2023, doi: 10.1016/j.cie.2023.109262.

[8] Á. Madridano, A. Al-Kaff, D. Martín, and A. De La Escalera, "Trajectory planning for multi-robot systems: Methods and applications," Expert Systems with Applications, vol. 173, p. 114660, 2021, doi: 10.1016/j.eswa.2021.114660.

[9] A. Marashian and A. Razminia, "Mobile robot's path-planning and path-tracking in static and dynamic environments: Dynamic programming approach," Robotics and Autonomous Systems, vol. 172, p. 104592, 2024, doi: https://doi.org/10.1016/j.robot.2023.104592.

[10] X. Li, L. Wang, Y. An, Q.-L. Huang, Y.-H. Cui, and H.-S. Hu, "Dynamic path planning of mobile robots using adaptive dynamic programming," Expert Systems with Applications, vol. 235, p. 121112, 2024, doi: 10.1016/j.eswa.2023.121112.

[11] C. Vignesh, M. Uma, and P. Sethuramalingam, "Development of rapidly exploring random tree based autonomous mobile robot navigation and velocity predictions using K-nearest neighbors with fuzzy logic analysis," International Journal on Interactive Design and Manufacturing (IJIDeM), pp. 1-25, 2024, doi: https://doi.org/10.1007/s12008-023-01701-1.

[12] C. Ntakolia, S. Moustakidis, and A. Siouras, "Autonomous path planning with obstacle avoidance for smart assistive systems," Expert Systems with Applications, vol. 213, p. 119049, 2023, doi: https://doi.org/10.1016/j.eswa.2022.119049.

[13] D. R. Parhi, "Chaos-based optimal path planning of humanoid robot using hybridized regression-gravity search algorithm in static and dynamic terrains," Applied Soft Computing, vol. 140, p. 110236, 2023, doi: https://doi.org/10.1016/j.asoc.2023.110236.

[14] J. Tang and H. Ma, "Mixed Integer Programming for Time-Optimal Multi-Robot Coverage Path Planning with Heuristics," IEEE Robotics and Automation Letters, 2023, doi: https://doi.org/10.1109/LRA.2023.3306996.

[15] T. Lei, T. Sellers, C. Luo, D. W. Carruth, and Z. Bi, "Graph-based robot optimal path planning with bio-inspired algorithms," Biomimetic Intelligence and Robotics, vol. 3, no. 3, p. 100119, 2023, doi: https://doi.org/10.1016/j.birob.2023.100119.

[16] E. García, J. R. Villar, Q. Tan, J. Sedano, and C. Chira, "An efficient multi-robot path planning solution using A* and coevolutionary algorithms," Integrated Computer-Aided Engineering, vol. 30, no. 1, pp. 41-52, 2023, doi: 10.3233/ICA-220695.

[17] Z. Ahmadirad, "Evaluating the influence of AI on market values in finance: distinguishing between authentic growth and speculative hype," International Journal of Advanced Research in Humanities and Law, vol. 1, no. 2, pp. 50-57, 2024, doi: https://doi.org/10.63053/ijrel.11.

[18] S. P. Rajput et al., "Using machine learning architecture to optimize and model the treatment process for saline water level analysis," Journal of Water Reuse and Desalination, 2022, doi: https://doi.org/10.2166/wrd.2022.069.

[19] M. Aghamohammadghasem, J. Azucena, F. Hashemian, H. Liao, S. Zhang, and H. Nachtmann, "System simulation and machine learning-based maintenance optimization for an inland waterway transportation system," in 2023 Winter Simulation Conference (WSC), 2023: IEEE, pp. 267-278, doi: https://doi.org/10.1109/WSC60868.2023.10408112.

[20] K. Xu, J. Lyu, and S. Manoochehri, "In situ process monitoring using acoustic emission and laser scanning techniques based on machine learning models," Journal of Manufacturing Processes, vol. 84, pp. 357-374, 2022, doi: https://doi.org/10.1016/j.jmapro.2022.10.002.

[21] R. Choupanzadeh and A. Zadehgol, "A Deep Neural Network Modeling Methodology for Efficient EMC Assessment of Shielding Enclosures Using MECA-Generated RCS Training Data," IEEE Transactions on Electromagnetic Compatibility, 2023, doi: https://doi.org/10.1109/TEMC.2023.3316916.

[22] M. Hajihosseinlou, A. Maghsoudi, and R. Ghezelbash, "A comprehensive evaluation of OPTICS, GMM and K-means clustering methodologies for geochemical anomaly detection connected with sample catchment basins," Geochemistry, p. 126094, 2024, doi: https://doi.org/10.1016/j.chemer.2024.126094.

[23] S. Vairachilai, A. Bostani, A. Mehbodniya, J. L. Webber, O. Hemakesavulu, and P. Vijayakumar, "Body Sensor 5 G Networks Utilising Deep Learning Architectures for Emotion Detection Based On EEG Signal Processing," Optik, p. 170469, 2022, doi: https://doi.org/10.1016/j.ijleo.2022.170469.

[24] A. Dutta, N. Masrourisaadat, and T. T. Doan, "Convergence Rates of Decentralized Gradient Dynamics over Cluster Networks: Multiple-Time-Scale Lyapunov Approach," in 2022 IEEE 61st Conference on Decision and Control (CDC), 2022: IEEE, pp. 6497-6502, doi: https://doi.org/10.1109/CDC51059.2022.9992900.

[25] S. R. Abdul Samad et al., "Analysis of the Performance Impact of Fine-Tuned Machine Learning Model for Phishing URL Detection," Electronics, vol. 12, no. 7, p. 1642, 2023, doi: https://doi.org/10.3390/electronics12071642.

[26] A. Larijani and F. Dehghani, "A Computationally Efficient Method for Increasing Confidentiality in Smart Electricity Networks," Electronics, vol. 13, no. 1, p. 170, 2023, doi: https://doi.org/10.3390/electronics13010170.

[27] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51-67, 2016, doi: https://doi.org/10.1016/j.advengsoft.2016.01.008.

[28] E. Bojnordi, S. J. Mousavirad, M. Pedram, G. Schaefer, and D. Oliva, "Improving the generalisation ability of neural networks using a Lévy flight distribution algorithm for classification problems," New Generation Computing, vol. 41, no. 2, pp. 225-242, 2023, doi: https://doi.org/10.1007/s00354-023-00214-5.

[29] K. V. Price, R. M. Storn, and J. A. Lampinen, "Benchmarking differential evolution," Differential Evolution: A Practical Approach to Global Optimization, pp. 135-187, 2005, doi: https://doi.org/10.1007/3-540-31306-0_3.

[30] Q. Yang et al., "Triple competitive differential evolution for global numerical optimization," Swarm and Evolutionary Computation, vol. 84, p. 101450, 2024, doi: https://doi.org/10.1016/j.swevo.2023.101450.

[31] F. H. Ajeil, I. K. Ibraheem, M. A. Sahib, and A. J. Humaidi, "Multi-objective path planning of an autonomous mobile robot using hybrid PSO-MFB optimization algorithm," Applied Soft Computing, vol. 89, p. 106076, 2020, doi: https://doi.org/10.1016/j.asoc.2020.106076.

[32] S. Wen, Z. Wen, D. Zhang, H. Zhang, and T. Wang, "A multi-robot path-planning algorithm for autonomous navigation using meta-reinforcement learning based on transfer learning," Applied Soft Computing, vol. 110, p. 107605, 2021, doi: https://doi.org/10.1016/j.asoc.2021.107605.

[33] P. Das and P. K. Jena, "Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators," Applied Soft Computing, vol. 92, p. 106312, 2020, doi: https://doi.org/10.1016/j.asoc.2020.106312.

[34] K. Suresh, R. Venkatesan, and S. Venugopal, "Mobile robot path planning using multi-objective genetic algorithm in industrial automation," Soft Computing, vol. 26, no. 15, pp. 7387-7400, 2022, doi: https://doi.org/10.1007/s00500-022-07300-8.

[35] A. Zou, L. Wang, W. Li, J. Cai, H. Wang, and T. Tan, "Mobile robot path planning using improved mayfly optimization algorithm and dynamic window approach," The Journal of Supercomputing, vol. 79, no. 8, pp. 8340-8367, 2023, doi: https://doi.org/10.1007/s11227-022-04998-z.

[36] Y. Cui, W. Hu, and A. Rahmani, "Multi-robot path planning using learning-based artificial bee colony algorithm," Engineering Applications of Artificial Intelligence, vol. 129, p. 107579, 2024.