IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 January 2024, accepted 15 February 2024, date of publication 19 February 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3367115

RESEARCH ARTICLE

A Hybrid HHO-AVOA for Path Planning of a Differential Wheeled Mobile Robot in Static and Dynamic Environments

ANBALAGAN LOGANATHAN AND NUR SYAZREEN AHMAD, (Member, IEEE)
School of Electrical and Electronic Engineering, Universiti Sains Malaysia, Nibong Tebal, Penang 14300, Malaysia

Corresponding author: Nur Syazreen Ahmad (syazreen@usm.my)

This work was supported by the Ministry of Higher Education Malaysia for Fundamental Research Grant Scheme under Project FRGS/1/2021/TK0/USM/02/18.

ABSTRACT This study presents a hybrid HHO-AVOA which is a novel optimization method that combines the strengths of Harris Hawks Optimization (HHO) and African Vulture Optimization Algorithm (AVOA) to address the path planning challenges encountered by differential wheeled mobile robots (DWMRs) navigating both static and dynamic environments, while accommodating kinematic constraints. By synergizing the strengths of both algorithms, the proposed hybrid method effectively mitigates the limitations of individual approaches, resulting in efficient and obstacle-avoiding navigation towards the target within reduced timeframes. To evaluate its efficiency, the proposed approach is compared against HHO and AVOA as well as other established methods which include whale optimization, grey wolf optimization and sine-cosine algorithms. Simulation results along with Monte Carlo analysis consistently demonstrate the superior performance of the hybrid method in both environments. In static scenarios, the hybrid algorithm achieves an average reduction of approximately 14% in path length and a 17% decrease in DWMR travel duration. In dynamic cases, it outperforms the rest with an average reduction of 27.6% in path length and a 27.2% decrease in travel duration. The algorithm's low computational complexity is also exhibited via its fast convergence during path optimization which is a crucial attribute for real-time implementation, particularly in dynamically changing environments that demand quick decision-making. The superiority of the proposed hybrid method to balance the exploration and exploitation is also affirmed through a Wilcoxon rank-sum test with a 95% confidence interval.

INDEX TERMS Dynamic environment, Harris Hawks optimization, hybrid, mobile robot, path planning, SCA, WOA.

I. INTRODUCTION

The adoption of autonomous mobile robots, particularly those employing differential drive configurations, has experienced a significant upsurge in recent years [1], [2]. This trend can be attributed, in part, to changing global dynamics that extend beyond the COVID-19 pandemic. Various sectors, including but not limited to healthcare, security patrols, and the food industry, have increasingly shifted their focus toward reducing human-to-human interactions [3]. Instead, they are embracing human-to-machine interactions to enhance safety and efficiency. Differential wheeled mobile robots (DWMRs) are well-suited to this transition due to their ability to navigate with precision and autonomy [4], [5].

In the domain of autonomous mobile robots, path planning emerges as a critical task, encompassing the determination of the optimal route while addressing performance objectives such as minimizing travel duration, distance, or energy consumption [6]. The extensive body of literature in this domain emphasizes the complexity of path planning for mobile robots and underscores several crucial challenges. One of these challenges is the imperative to generate routes that are not only collision-free but also well-structured, ensuring feasible navigation for mobile robots

The associate editor coordinating the review of this manuscript and approving it for publication was Mou Chen.

particularly those that are subject to kinematic and dynamic constraints [7], [8], [9], [10].

Addressing these challenges has led to a variety of approaches. Classical techniques like the Roadmap Approach [11], Cell Decomposition [12], and Artificial Potential Field [13] have historically prevailed due to their effectiveness in solving path planning problems [14]. However, they come with certain limitations, including computational complexity in dense environments, rigidity in accommodating additional constraints or objectives, suboptimal performance in dynamic settings, and lack of robustness. To address these limitations, various artificial intelligence-based approaches have been proposed, including neural dynamics [15], adaptive neuro-fuzzy inference system [16], Ant Colony Optimization(ACO) [17], Particle Swarm Optimization (PSO) [18], Artificial Bee Colony (ABC) [19], Grey Wolf Optimization (GWO) [20], [21], and Whale Optimization algorithm (WOA) [22], [23]. These methods draw inspiration from the collective behavior of natural systems, such as ants, birds, bees, whales and wolves, to solve complex problems [24].

While the direct application of the aforementioned algorithms to robot path planning has shown promise, they come with inherent problems and limitations, often revolving around the delicate balance between exploration and exploitation phases in the quest for optimal paths [25]. Consequently, many researchers have turned to modifying these algorithms to achieve improved results [26], [27], [28], [29], [30]. For instance, in [31], an enhanced WOA is presented, addressing the original algorithm's deficiencies in indoor robot path planning by incorporating improved chaotic mapping for population initialization and employing fused Corsi variance and weighting strategies to enhance path quality. Another approach by fusing WOA and fuzzy logic (FL) is introduced in [32] which leads to a 20.63% improvement in path length optimization. In [33], an enhanced variant of the sine-cosine algorithm (SCA) is introduced to address multi-robot path planning challenges in complex environments featuring both static and dynamic obstacles.

The Harris Hawks Optimization (HHO) algorithm, introduced in 2019, draws inspiration from hawks' collaborative hunting strategies, where they surprise their prey through coordinated attacks and adapt to prey movements [34]. In the context of static environment path planning, as demonstrated in [35], HHO exhibits faster computation speeds compared to WOA and SCA. Moreover, a fusion of HHO and the Dijkstra method, as presented in [36], outperforms HHO alone in static environments. Further enhancements to HHO are introduced in [37], addressing issues like out-of-bounds rates and precision by incorporating a circle map for initial population generation and a nonlinear jump strength mechanism, proving its efficiency in terms of accuracy, and convergence speed, especially in 2D grid map path planning applications.

Another significant challenge in path planning is the algorithm's capacity to adjust to uncertain or dynamic environments [38], [39], [40], [41], [42]. Vega-Brown and Roy [43] for instance, introduced a factored orbital Bellman tree technique aimed at manipulating movable obstacles to facilitate the creation of the shortest path for a manipulator. In environments featuring dynamic non-movable obstacles, generating the shortest path requires identifying uncertain zones, but this may pose a disadvantage due to the extensive computation needed for optimal path generation [44]. In such cases, the necessity for local planning with rapid computation becomes crucial to ensure timely goal attainment for the robot. In the work by Hasankhani et al. [45], the path planner is crafted using reinforcement learning to tackle the highly nonlinear and uncertain oceanic environment. Nakrani and Joshi [46] introduced a fuzzy-based obstacle avoidance controller with ulstrasonic sensor outputs as inputs to the FL to accomplish autonomous parallel parking in the presence of static and dynamic obstacles. In [47], the primary emphasis is on addressing the challenges of longitudinal and lateral control coupling to ensure safety and stability in collision avoidance scenarios. Another study referenced in [48] leverages the African Vulture Optimization Algorithm (AVOA), which was a new metaheuristic algorithm introduced in 2021 by Abdollahzadeh et al. [49], inspired by the foraging and navigation behaviors of African vultures. Simulation results demonstrate AVOA's exceptional proficiency in obstacle avoidance, even in challenging local minima scenarios. Furthermore, in terms of improving path length, AVOA surpasses PSO by 2.21% and the fuzzy-based method by 1.02% [50].

While the aforementioned techniques have displayed potential in enhancing path planning strategies, addressing dynamic obstacles continues to pose challenges, primarily because of the necessity to incorporate computation time as an additional constraint to optimize the route. The main contributions of this study are summarized as follows:

- A new optimization method based on hybrid HHO-AVOA is proposed for path planning of a DWMR operating within both static and dynamic environments while accommodating kinematic constraints. The superiority of the proposed HHO-AVOA algorithm to balance the exploration and exploitation is verified via Wilcoxon rank-sum test with a 95% confidence interval.
- Simulation results along with Monte Carlo analysis comparing the proposed algorithm to its individual components, i.e. HHO [35] and AVOA [49]), as well as other state-of-the-art algorithms (i.e. WOA, GWO and SCA) demonstrate the significant performance improvement introduced by the proposed hybrid method. In the static case, it achieves an average reduction of approximately 14% in path length and a 17% decrease in DWMR travel duration. In the dynamic case, the proposed algorithm exhibits remarkable

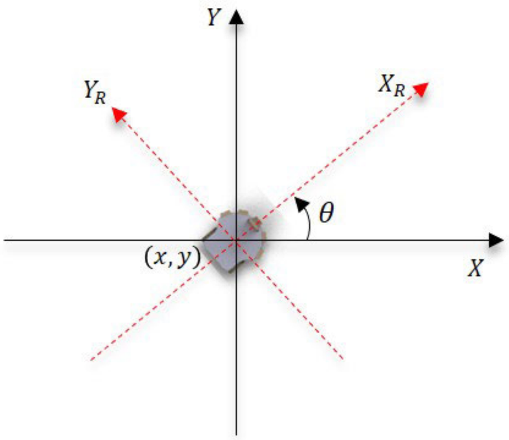

FIGURE 1. Portrayal of the DWMR in $[X, Y]$ and $[X_R, Y_R]$ planes.

superiority over its strongest competitor, with an average reduction of 27.6% in path length and a 27.2% decrease in travel duration.
- In both environments, the proposed algorithm's low computational complexity is evident through its fast convergence during path optimization. This attribute is essential for real-time implementation where rapid computational speed is crucial, particularly in dynamically changing environments that demand quick decision-making.

The remainder of the paper is organized as follows: Section II presents the methodology, covering DWMR modeling, the HHO algorithm, the AVOA algorithm, and the proposed hybrid HHO-AVOA approach specifically tailored for DWMR path planning. Section III presents a thorough examination of simulation results accompanied by Monte Carlo analysis, covering scenarios in both static and dynamic environments. Section IV concludes the paper by providing a summary of the study's key findings and contributions, and outlining potential directions for future research.

II. METHODOLOGY

A. DIFFERENTIAL WHEELED MOBILE ROBOT (DWMR)

In odometry, the robot's current position relies on the measurement of change from the robot's known starting position. The odometry-based localization is based on robot wheels' movement data collected by motion sensors to estimate the change of position over time. In this work, the estimation of the path taken and the heading direction of the mobile robot is done by using the data from the encoders on both wheels. Fig. 1 depicts the position of robot in relation to two different frames, namely $[X, Y]$ and $[X_R, Y_R]$. $[X, Y]$ represents the world coordinate system while $[X_R, Y_R]$ denotes the robot-attached frame [51]. Both linear velocity, v as well as angular velocity, ω of the robot can be represented by the following equations:

$$v = \frac{(\omega_r + \omega_l)r}{2}, \quad \omega = \frac{(\omega_r - \omega_l)r}{D}, \tag{1}$$

where r is wheel's radius, D is the azimuth length between the wheels, and ω_r and ω_l represent the right and left wheels' angular velocities respectively. Let (x_0, y_0, θ_0) denote to the robot's previous position. When the robot is moving forward along the X_R-axis, only the linear velocity v is non-zero, and $\theta = \theta_0$. Consequently, we have $\dot{x} = v\cos\theta_0$ and $\dot{y} = v\sin\theta_0$. The robot's current position can be obtained by integrating $\dot{x}$ and $\dot{y}$, resulting in:

$$x = vt_s \cos\theta_0 + x_0,$$
$$y = vt_s \sin\theta_0 + y_0,$$

where t_s is the sampling time. Due to nonholonomic constraints, the robot is unable to move sideways. Thus, when the robot steers, the trajectory is a circular arc with a radius $\rho = v/\omega$. The current heading angle of the robot can be calculated as $\theta = \int_0^{t_s} \omega dt$. The position of the robot can therefore be estimated with

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \rho\sin(\omega t_s) \\ \rho(1-\cos(\omega t_s)) \\ \omega t_s \end{bmatrix} + \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix}.$$

In this study, we examine a DWMR operating within specific constraints, where the maximum angular velocity is set at 3 rad/s, and the maximum linear velocity is limited to 0.5 m/s. Consequently, paths of comparable lengths but with distinct shapes may not yield identical travel durations due to these imposed restrictions. Notably, paths featuring more curves necessitate additional turning, resulting in slightly longer durations.

B. HARRIS HAWKS OPTIMIZATION (HHO)

The Harris Hawks Optimization (HHO) algorithm mimics how hawks hunt prey to find the best solution for a problem. It starts with an initial guess and then continually refines it through two phases: exploring broadly and exploiting locally. It uses energy as a guide during the search, and the switch between global exploration and local exploitation depends on the prey's energy equation. The mathematical expression for the energy-based search is as follows:

$$E_0 = 2\epsilon - 1 \tag{2}$$

$$E = 2E_0\big(1 - \frac{k}{k_{max}}\big); \tag{3}$$

where E denotes the escape energy of the rabbit, $E_0 \in [-1, 1]$ denotes its initial energy, and k_{max} is the total number of iterations. If $E \geq 1$, HHO initiates the global exploration phase, where it searches extensively across the search space for a solution. Conversely, when $E < 1$, HHO transitions into the local exploitation phase. In this phase, it focuses on refining and enhancing a solution previously discovered within a more restricted area of the search space.

In the global exploration phase, the Harris Hawks algorithm explores the search space to find a solution. It does this by conducting random inspections and surveillance of the area, along with employing two strategies for random prey searching. The position is updated during each iteration based on the probability of "q" using the following equations:

$$X(k+1) \tag{4}$$

$$= \begin{cases} X_{ran}(k) - r_1|X_{ran}(k) - 2r_2X(k)| & \text{if } q \geq 0.5 \\ X_{rab}(k) - X_m(k) - r_3|lb + r_4(ub - lb)| & \text{if } q < 0.5 \end{cases} \tag{5}$$

where $X(k+1)$ denotes the position of the hawks in the next iteration, $X_{rab}(k)$ is the current position of the rabbit, $X(k)$ is the current position vector of hawks, the variables $r_j, j = 1, \ldots 4$ and q are random numbers ranging from 0 to 1, with values updated in each iteration. Additionally, "lb" and "ub" denote the lower and upper bounds of the search space respectively. $X_{ran}(k)$ represents a randomly chosen hawk from the current population, while $X_m(k)$ represents the average position of the current population of hawks. This average position is calculated using the following formula:

$$X_m(k) = \frac{1}{N}\sum_{i=1}^{N} X_i(k) \tag{6}$$

where $X_i(k)$ represents the position of i-th hawks in the k-th iteration, and N signifies the total number of hawks.

In the local exploitation phase, the HHO algorithm actively engages in improving a previously located rabbit. During this process, the rabbit may attempt to escape, and the energy level, E, is utilized to decide the most effective strategy for the hawks to pursue the rabbit. Furthermore, the probability of successfully capturing the escaping prey, denoted by r which is randomly generated during initialization, influences the choice of the optimal approach. To handle this scenario, the algorithm employs four distinct strategies, which are detailed below:

i. **Soft Besiege:** This behavior is represented by the following equations:

$$\begin{cases} X(k+1) = \Delta X(k) - E|JX_{rab}(k) - X(k)| \\ \Delta X(k) = X_{rab}(k) - X(k) \\ J = 2(1 - r_5) \end{cases} \tag{7}$$

where $\Delta X(k)$ represents the positional difference between the rabbit's current location and its initial position vector, $r_5 \in [0, 1]$ is a randomly generated number which is utilized to calculate the random jump strength of the rabbit during the escape procedure, and J denotes the stochastic nature of the rabbit's movements which undergoes random variations in each iteration, mimicking the inherent unpredictability of rabbit motion.

ii. **Hard Besiege:** At this stage, the rabbit's available energy for escaping is insufficient, leading the hawks to aggressively chase the rabbit. Consequently, the rabbit's position is updated according to the following equation:

$$X(k+1) = X_{rab}(k) - E|\Delta(k)| \tag{8}$$

iii. **Soft Besiege with progressive rapid dives:** The hawks will first begin with a soft besiege strategy when the escaping energy is deemed sufficient, before transitioning to an aggressive approach. The HHO algorithm integrates the Levy function to mimic the jumping behavior and escape tactics of the rabbit. To execute a soft besiege, it is assumed that the hawks can determine their next move according to the rule described by

$$A_1 = X_{rab}(k) - E|JX_{rab}(k) - X(k)| \tag{9}$$

It is also assumed that they will engage in a dive pattern based on the Levy-based patterns as follows

$$A_2 = Y + S \times Levy(D) \tag{10}$$

where D represents the problem dimension, S is a random vector with a size of $1 \times D$, and "$Levy$" is the Levy flight function computed using

$$Levy(x) = 0.01 \times \frac{u \times \sigma}{|v|^{1/\beta}}, \tag{11}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\frac{\pi\beta}{2})}{\Gamma(\frac{1+\beta}{2}) \times \beta \times 2^{(}\frac{\beta-1}{2})}\right)^{1/\beta} \tag{12}$$

This condition is formulated as follows:

$$X(k+1) = \begin{cases} A_1 & \text{if} \quad F(A_1) < F(X(k)) \\ A_2 & \text{if} \quad F(A_2) < F(X(k)) \end{cases} \tag{13}$$

where A_1 and A_2 are obtained from (9) and (10) respectively.

iv. **Hard Besiege with progressive rapid dives:** As the rabbit lacks the energy required for escape, the hawks will initiate a hard besiege before executing a sudden maneuver to capture the rabbit. This mathematical movement is represented as:

$$X(k+1) = \begin{cases} A_1 & \text{if} \quad F(A_1) < F(X(k)) \\ A_2 & \text{if} \quad F(A_2) < F(X(k)) \end{cases} \tag{14}$$

where A_1 and A_2 can be computed as follows

$$A_1 = X_{rab}(k) - E|JX_{rab}(k) - X_m(k)| \tag{15}$$

$$A_2 = Y + S \times Levy(D) \tag{16}$$

The pseudocode for the standard HHO algorithm is presented in Alg. 1

C. AFRICAN VULTURE OPTIMIZATION ALGORITHM (AVOA)

The African Vulture Optimization Algorithm (AVOA) is a versatile and easy-to-implement meta-heuristic method inspired by African vultures' eating and orientation behaviors [49]. AVOA is widely used across various optimization domains, offering flexibility for adjustments and applications that lead to optimal results. Vultures in the wild exhibit distinct behaviors based on their physical capabilities,

Algorithm 1 HHO Algorithm

```
Initialize hawks population with random positions, $X_i (i = 1, \ldots, N)$
while $k \leq k_{max}$ do
    Evaluate the fitness of each hawk
    Let $X_{rab}(k)$ be the best position of the rabbit
    for $i = 1, \ldots, N$ do
        Update $E_0$ and $J$ based on (2)
        Update $E$ based on (3)
        if $E \geq 1$ then
            Update $X$ according to (4)
        else if $E < 1$ then
            if $E \geq 0.5$ and $u \geq 0.5$ then
                Update $X$ according to (7)
            else if $E < 0.5$ and $u \geq 0.5$ then
                Update $X$ according to (8)
            else if $E \geq 0.5$ and $u < 0.5$ then
                Update $X$ according to (13)
            else if $E < 0.5$ and $u < 0.5$ then
                Update $X$ according to (14)
            end if
        end if
    end for
    $k++$
end while
Return $X_{rab}$
```

falling into two groups. They are adept at evading traps due to their insatiable hunger and persistent hunting for food. In the algorithm, the two strongest and most efficient vultures are considered as the most robust and optimal solutions. The following subsections will outline the algorithm's general steps.

1) DETERMINING THE BEST VULTURE IN ANY GROUP

Following the formation of the initial population, the fitness of all solutions is evaluated, and the top-performing solution is designated as the best vulture in the first group, denoted as BV_1, while the second-best solution is identified as the best vulture in the second group, denoted as BV_2. The corresponding rule is written as follows:

$$R(k) = \begin{cases} BV_1 & \text{if} \quad p_i = B_1 \\ BV_2 & \text{if} \quad p_i = B_2 \end{cases} \tag{17}$$

where B_1 and B_2 are quantified values determined prior to the search operation. These parameters have values ranging from zero to one, and $B_1 + B_2 = 1$. Additionally, the selection of one of the best solutions is achieved through a roulette wheel mechanism as follows:

$$p_i = F_i / (\sum_{i=1}^{n} F_i) \tag{18}$$

2) RATE OF STARVATION OF VULTURES

The frequent food-seeking behavior of vultures and their increased endurance after eating allows them to travel farther during their pursuit of food. They lose energy when they are hungry, making it harder for them to travel great distances in search of food. In such circumstances, they frequently seek food close to stronger vultures and may become combative out of hunger. This behavior can be modeled mathematically as follows:

$$t = h \times \left(\sin^w \left(\frac{\pi}{2} \times \frac{k}{k_{max}} \right) + \cos \left(\frac{\pi}{2} \times \frac{k}{k_{max}} \right) - 1 \right) \tag{19}$$

$$F = (2 \times r_{v1} + 1) \times S \times \left(1 - \frac{k}{k_{max}} \right) + t. \tag{20}$$

In (19) and (20), F represents the vultures' satiety level, $S \in [-1, 1]$, $h \in [-2, 2]$ and $r_{v1} \in [0, 1]$ are random numbers. When the value of S drops below 0, it signifies that the vulture is in a state of hunger, while an increase in S to 0 indicates satiety. In the AVOA algorithm, the variable F plays a pivotal role in determining whether the algorithm enters an exploration or exploitation phase. Specifically, if $|F| > 1$, the algorithm initiates the exploration phase, whereas if $|F| \leq 1$, it begins the exploitation phase.

In the exploration phase, the vultures employ one of two strategies to investigate random areas, and the selection of the strategy is governed by a predefined parameter called $P_1 \in [0, 1]$ which is determined prior to the commencement of the search operation. To decide which strategy to utilize during the "randP1" exploration phase, a random number is generated within the range of 0 to 1, denoted as r_{p1}. If $P_1 \geq r_{p1}$ parameter, then the following rule is applied

$$\begin{cases} VP(k+1) = R(k) - D(k) \times F \\ D(k) = |X \times R(k) - VP(k)| \end{cases} . \tag{21}$$

where $VP(k)$ and $VP(k+1)$ represent the vector positions of the vulture in the current and next iterations respectively, $R(i)$ denotes one of the best vultures, F is the satiation rate of the vulture, and X indicates the random movement of the vulture for safeguarding the food from other vultures. The random motion, which varies with each iteration and is calculated using the formula $X = 2 \times rand$, where $rand \in [0, 1]$ is a random number that acts as the amplifying coefficient of X. In the case where $P_1 < r_{p1}$, then the following equation is applied

$$VP(k+1) = R(k) - F + r_{v2} \times ((ub - lb) \times r_{v3} + lb) \tag{22}$$

where $r_{v2}, r_{v3} \in [0, 1]$ are random numbers, and "ub" and "lb" represent the lower bound and lower bound of the variables respectively.

During the exploitation stage, there are two main phases involved, namely Phase 1 and Phase 2. The AVOA transitions to Phase 1 when $|F| \in [0.5, 1)$. This phase consists of two strategies called rotating flight and siege-flight strategies, which are detailed below:

a: SIEGE FLIGHT

The fierce rivalry for food acquisition between weaker and stronger vultures is simulated through the following rule:

$$\begin{cases} VP(k+1) = D(k) \times (F + r_{v4}) - d(t) \\ d(t) = R(k) - VP(k) \end{cases} \tag{23}$$

where $r_4 \in [0, 1]$ is a random number used to amplify the random coefficient. The equation gives the distance between one of the best vultures of the two groups and the vulture.

b: ROTATING FLIGHT

The rotational flight patterns of the vultures can be modeled mathematically into a spiral motion. With this approach, an equation involving all the vultures and one of the two best vultures is created. This movement can be expressed mathematically as follows:

$$\begin{cases} S_1 = R(k) \times \left(\frac{r_{v5} \times VP(k)}{2\pi}\right) \times \cos(P(k)) \\ S_2 = R(k) \times \left(\frac{r_{v6} \times VP(k)}{2\pi}\right) \times \cos(P(k)) \\ VP(k+1) = R(i) - (S_1 + S_2) \end{cases} \tag{24}$$

where $r_{v5}, r_{v6} \in [0, 1]$ are random numbers.

To select the strategy, a variable $P_2 \in [0, 1]$ is introduced before the search operation begins. A random number, r_{p2}, is generated at the onset of this phase. Based on r_{p2} and P_2, the selection of strategy is formulated as follows

$$VP(k+1) = \begin{cases} (23) & \text{if} \quad P_2 \geq r_{p2} \\ (24) & \text{if} \quad P_2 < r_{p2} \end{cases} \tag{25}$$

The AVOA transitions to Phase 2 when $|F| < 0.5$. This phase consists of two strategies, namely "Congregation of various types of vultures around the food source" and "Aggressive competition for food". These strategies are described below:

c: CONGREGATION OF VARIOUS TYPES OF VULTURES AROUND THE FOOD SOURCE

In this stage of AVOA, the vultures' movement to gather around the food source is observed. When vultures are starving, there can occasionally be fierce competition for food, which can cause several different kinds of vultures to congregate around a single food source which can be modeled as follows:

$$\begin{cases} A_1 = BV_1(k) - \frac{BV_1(k) \times VP(k)}{BV_1(k) - VP(k)^2} \times F \\ A_2 = BV_2(k) - \frac{BV_2(k) \times VP(k)}{BV_2(k) - VP(k)^2} \times F \\ VP(k+1) = \frac{A_1 + A_2}{2} \end{cases} \tag{26}$$

where BV_1 and BV_2 are the best vulture from the first group and second group respectively.

d: AGGRESSIVE COMPETITION FOR FOOD

At this stage of AVOA, the leading vultures become weak and famished, thus unable to compete with other vultures for food. While other vultures which are starving will become aggressive and start to move toward the leading vulture. Their movement can be expressed as

$$VP(k+1) = R(k) - |d(t)| \times F \times Levy(d) \tag{27}$$

where $d(t)$ represents the distance between a vulture and one of the best vultures of both groups. The usage of the Levy flight function increases the efficacy of the AVOA algorithm.

To select the strategy, a variable $P_3 \in [0, 1]$ is introduced before the search begins. A random number, r_{p3}, is generated at the onset of this phase. Based on r_{p3} and P_3, the selection of strategy is formulated as follows

$$VP(k+1) = \begin{cases} (26) & \text{if} \quad P_3 \geq r_{p3} \\ (27) & \text{if} \quad P_3 < r_{p3}. \end{cases} \tag{28}$$

The pseudocode for the standard AVOA algorithm is presented in Alg. 2.

D. PROPOSED HYBRID HHO-AVOA FOR PATH PLANNING OF DWMR

To improve the efficiency of path planning for a DWMR, we introduce a hybrid algorithm called HHO-AVOA, which combines the HHO and AVOA algorithms. In our proposed approach, we leverage the AVOA algorithm for global exploration, and when the value of F falls below 1, the algorithm seamlessly transitions to the HHO strategy for local exploitation.

In the field of robot path planning, the primary focus is on minimizing the path length when determining a robot's route. This is crucial because the path length has a direct impact on both the robot's energy consumption and travel duration. The ultimate goal is to identify the shortest feasible route while effectively avoiding obstacles. In our notation, we represent the path length as q, which is computed as the cumulative length of all individual segments comprising the path as follows:

$$f_L(q) = \sum_{i=0}^{n-1} |q_i, q_{i+1}| \tag{29}$$

where $|q_i, q_{i+1}|$ denotes the distance between two adjacent points. Another objective in path planning is to ensure path safety. In this work, the obstacles are assumed to be circular. This avoidance strategy is vital as it safeguards the robot from potential damage and ensures a smoother path without abrupt alterations, ultimately leading to more efficient robot motion. The formulation for the obstacle avoidance rule is as

Algorithm 2 AVOA Algorithm

```
Initialize vultures population with random positions, $VP_i(i = 1, \ldots, N)$
while $k \leq k_{max}$ do
    Evaluate the fitness of each vulture
    Set $VP_{bv1}(k)$ as the location of the best vulture from Group 1
    Set $VP_{bv2}(k)$ as the location of the best vulture from Group 2
    for $i = 1, \ldots, N$ do
        Select $R$ using (17)
        Update $F$ based on (20)
        if $|F| \geq 1$ then
            if $P_1 \geq r_{p1}$ then
                Update $VP$ according to (21)
            else
                Update $VP$ according to (22)
            end if
        else if $|F| < 1$ then
            if $|F| < 0.5$ then
                if $P_2 \geq r_{p2}$ then
                    Update $VP$ according to (23)
                else
                    Update $VP$ according to (24)
                end if
            else
                if $P_3 \geq r_{p3}$ then
                    Update $VP$ according to (26)
                else
                    Update $VP$ according to (27)
                end if
            end if
        end if
    end for
    $k++$
end while
Return $VP_{bv1}$
```

follows:

$$\begin{cases} d(i) = \sqrt{(x - x_{oi})^2 + (y - y_{oi})^2} \\ \gamma(i) = \max\left(1 - \dfrac{d(i)}{R_{obs}(i)}, 0\right) \\ v_L(\gamma) = \dfrac{1}{n_o} \displaystyle\sum_{i=1}^{n_o} \gamma(i) \end{cases} \tag{30}$$

where $d(i)$ represents the distance between the DWMR and the center of the i-th obstacle, denoted as (x_{oi}, y_{oi}), n_o is the total number of obstacles, and $R_{obs}(i)$ is the radius of the i-th obstacle. If the magnitude of the v_L is more than zero, the robot has entered the obstacles' region. Thus, to optimize the search for the shortest route while ensuring obstacle avoidance, the cost function can be expressed as follows:

$$f = f_L \times (1 + \mu v_L). \tag{31}$$

where μ refers to a penalty constant. This formulation yields a higher value when the DWMR enters a hazardous zone close to obstacles, effectively penalizing such occurrences.

In addition, as elaborated in Section II-A, equivalent path lengths for a DWMR may not necessarily correspond to similar travel durations. Thus, the search for the optimal route also encompasses seeking an optimal travel duration,wherein the goal is to prevent an increase in duration after each iteration. The travel duration of the DWMR is computed using the formula:

$$T_f = \sum_{j=1}^{n_w - 1} \frac{\sqrt{(x_j - x_{j+1})^2 + (y_j - y_{j+1})^2}}{v} + \sum_{j=1}^{n_w - 1} \frac{\theta_{j+1} - \theta_j}{\omega} \tag{32}$$

where n_w refers to the total number of waypoints, v is the DWMR's linear velocity, ω is the DWMR's angular velocity, and (x_j, y_j, θ_j) and $(x_{j+1}, y_{j+1}, \theta_{j+1})$ are coordinates of adjacent waypoints. This formula eliminates the requirement for simulating the DWMR before recording the duration, thereby avoiding computational intensity during the path optimization.

Furthermore, the assessment of the proposed method, particularly in dynamic environments, is contingent on addressing the computational burden. Hence, the maximum computation time is considered as the stopping condition. The detailed representation of the proposed algorithm is provided in Alg. 3. The combination of AVOA-based exploration and HHO-based exploitation can improve the path planning strategy of a DWMR in both static and dynamic environments as demonstrated in the next section.

To further evaluate the proposed algorithm, the Wilcoxon rank-sum statistical test with a 5% significance level is conducted. This involves forming two hypotheses: the null hypothesis assumes no significant difference between the two groups, while the alternative hypothesis challenges this assumption. The test begins by separately grouping the generated waypoints of both the hybrid and comparison algorithms. These waypoints are then combined and ranked in increasing order. The rank sum of each group is calculated and the minimum value, U between both groups is recorded. Once the sample sizes, n_1 and n_2 are calculated for both groups, the expected value of U under the null hypothesis is calculated as follows:

$$E(U) = \frac{n_1 \times n_2}{2}.$$

Next, the standard deviation, $\sigma(U)$, for the null hypothesis and the z-score, are computed using the following equations

$$\sigma(U) = \sqrt{\frac{n_1 \times n_2 \times (n_1 + n_2 + 1)}{12}}, \quad z = \frac{U - E(U)}{\sigma(U)}.$$

By using the value of z obtained above, the cumulative probability in a standard normal distribution is recorded and the two-tailed p-value is calculated by mutiplying the probability obtained with 2. In the context of path planning,

Algorithm 3 Proposed Hybrid HHO-AVOA Algorithm

```
Initialize vultures population with random positions, $VP_i(i = 1, \ldots, N)$
Initialize hawks population with random positions, $X_i(i = 1, \ldots, N)$
while stopping condition is not met do
    Evaluate the fitness of each vulture using (29)-(31)
    Set $VP_{bv1}(k)$ as the best location from first vulture group
    Set $VP_{bv2}(k)$ as the best location from second vulture group
    for $i = 1, \ldots, N$ do
        Select $R$ using (17)
        Update $F$ based on (20)
        if $|F| \geq 1$ then
            if $P_1 \geq r_{p1}$ then
                Update location according to (21)
            else
                Update location according to (22)
            end if
        else if $|F| < 1$ then
            if $E \geq 0.5$ and $u \geq 0.5$ then
                Update location according to (7)
            else if $E < 0.5$ and $u \geq 0.5$ then
                Update location according to (8)
            else if $E \geq 0.5$ and $u < 0.5$ then
                Update location according to (13)
            else if $E < 0.5$ and $u < 0.5$ then
                Update location according to (14)
            end if
        end if
    end for
end while
Set $X_{rab}$ as the best location
```

obtaining a low p-value below 0.05 is crucial as it indicates that the proposed algorithm dominates the other in terms of performance.

III. RESULTS AND DISCUSSIONS

This section presents the simulation results of the proposed hybrid HHO-AVOA in both static and dynamic environments. In these settings, different-sized obstacles are positioned at coordinates (x_{oi}, y_{oi}), where i denotes each specific obstacle. The corresponding coordinates are detailed in Table 1. The simulations are executed using MATLAB R2022a on an Intel(R) Xeon processor with 64-bit architecture and 8GB of RAM. For each scenario, both the length of the optimal path and the time taken for the DWMR to traverse the path are recorded. Table 2 provides the list of the parameter settings utilized for all the path planning algorithms.

To demonstrate the effectiveness of the proposed method, it is compared against its constituent components, namely, the AVOA [48] and HHO [35] algorithms, both of which have been recently introduced in the literature for path planning in static and dynamic environments. Furthermore, we extended our comparisons to other well-established algorithms in the field of path planning, specifically WOA, GWO, and SCA. A comprehensive Monte Carlo analysis, comprising 50 trials with diverse initial conditions, including varying start and goal positions, is conducted to ensure a fair assessment among the algorithms as well as to demonstrate the robustness of each algorithm. This analysis also serves to reinforce the substantial improvement achieved by the proposed algorithm.

TABLE 1. Simulation setup for the static and dynamic environments.

	Obstacle's Coordinate (m)	
Obstacle	Static	Dynamic
(x_{o1}, y_{o1})	(2.5,0.0)	(-2.0,-4.0)
(x_{o2}, y_{o2})	(4.0,4.4)	(-2.0,2.0)
(x_{o3}, y_{o3})	(7.0,2.0)	(1.0,0.0)
(x_{o4}, y_{o4})	(1.0,3.0)	(-6.0,0.0)

TABLE 2. Parameter setting for all the algorithms.

Parameter	Value
Number of search agents, N	100
Maximum computation time (s)	10
Penalty constant, μ	1000
Number of handling points	3
Dimensions	6
Lower bound (m)	-10
Upper bound (m)	10

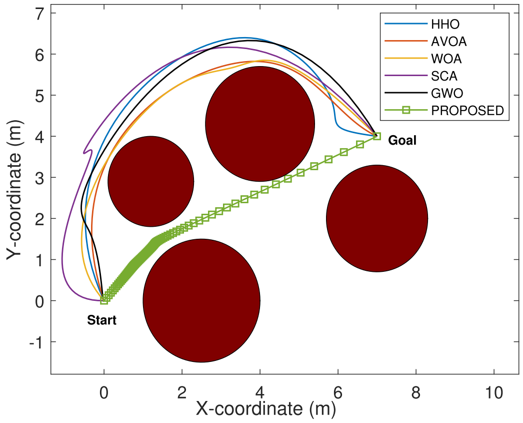

FIGURE 2. Optimal paths generated by the algorithms in the static environment from one of the trials.

Fig. 2 visualizes the optimal paths generated by the algorithms in the static environment from one of the trials where the positions of the obstacles, represented by maroon circular blocks, remain stationary. The corresponding convergence curves are displayed in Fig. 3 which clearly demonstrates that the proposed hybrid HHO-AVOA not only achieves the lowest best cost but also converges to this value in approximately 2 s, which is significantly faster than the other methods.

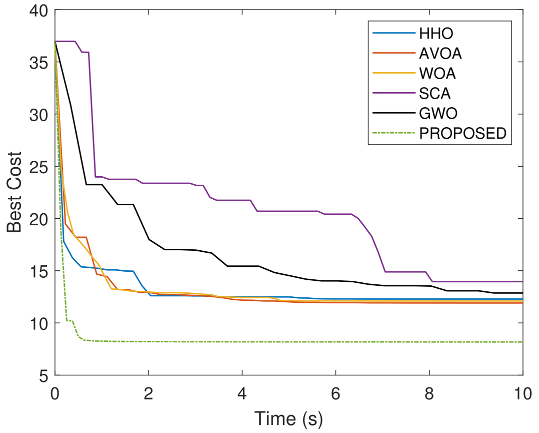

FIGURE 3. Convergence curves corresponding to the paths shown in Fig. 2. It is evident that the proposed method achieves a significantly faster convergence with the lowest best cost compared to the other methods.

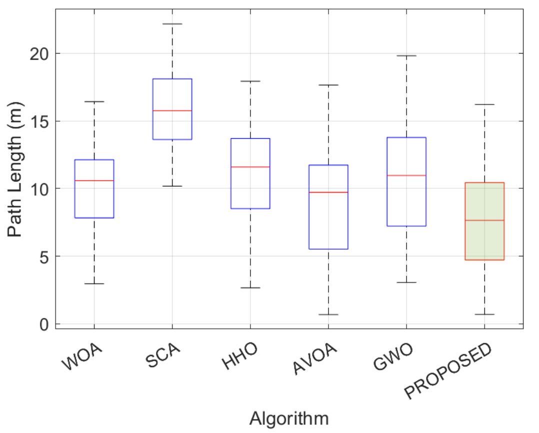

FIGURE 4. Box-plot depicting the comparison of path lengths obtained through the proposed hybrid HHO-AVOA and other algorithms across 50 trials in the static environment.

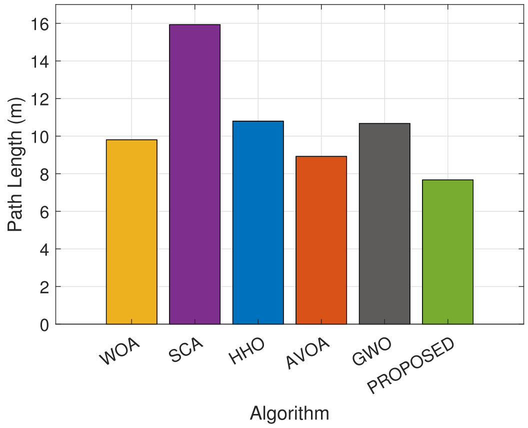

FIGURE 5. Comparison of average path lengths obtained through the proposed hybrid HHO-AVOA against other algorithms based on 50 trials in the static environment.

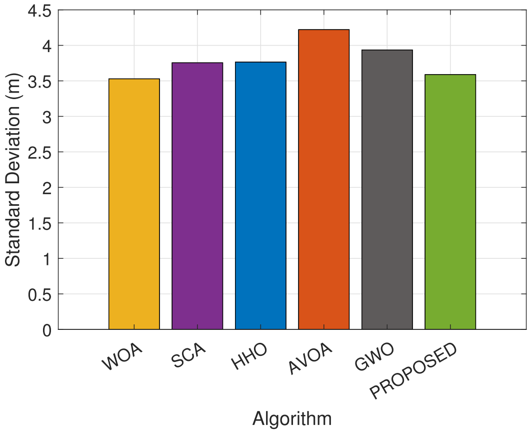

FIGURE 6. Comparison of standard deviations obtained through the proposed hybrid HHO-AVOA against other algorithms based on 50 trials in the static environment.

The outcomes of the Monte Carlo analysis are illustrated in Fig. 4 where the box plots compare the path lengths across 50 trials in the static environment, evaluating the proposed hybrid HHO-AVOA against other algorithms. While the interquartile range (IQR) of the proposed algorithm, represented by the green-colored box, is marginally larger than that of WOA, SCA, and HHO, it is noteworthy that its minimum and maximum values are considerably lower than theirs. Moreover, the proposed algorithm distinctly presents the lowest median value which is a crucial factor for optimal route generation in the context of path planning.

To better visualize the significant performance improvement of the proposed algorithm, Figs. 5 and 6 present another statistical analysis encompassing average path length and standard deviation, both derived from the 50 trials. By analyzing Fig. 5, it is evident that the proposed algorithm outperforms every other algorithm considered in the static environment. The consistency of each algorithm in producing the path is also verified by the standard deviation value recorded in Fig. 6 which is almost similar to all. Fig. 7 visualizes the time taken for the DWMR to traverse the paths generated by each algorithm in the static environment. Similar to the analysis in Fig. 4, the minimum and maximum values of the IQR, along with the median value resulting from the proposed method are significantly lower than those of the other methods. This result suggests that the proposed method is capable of minimizing both the path length and the travel duration of the DWMR.

With regard to the dynamic environment, the outcomes presented in Fig. 8 reveal a parallel pattern to the static case where the proposed method considerably outperforms the rest in terms of the shortest path generation. In this scenario, the positions of the obstacles in the figures represent their ultimate coordinates after undergoing random motion as indicated by the double-headed arrows, with middle positions changing within a 2-meter radius every 0.2 s. The corresponding convergence curves as shown in Fig. 9 further emphasize the superior performance of the hybrid HHO-AVOA approach as it only requires approximately 6 s to

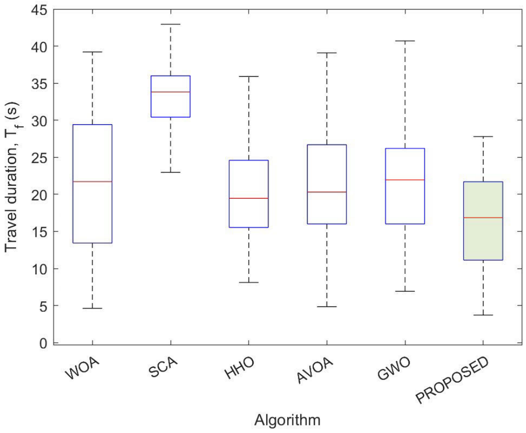

FIGURE 7. Box-plot depicting the comparison of travel durations obtained through the proposed hybrid HHO-AVOA and other algorithms across 50 trials in the static environment.

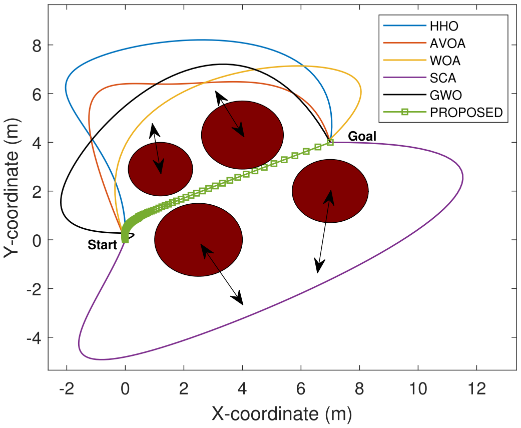

FIGURE 8. Optimal paths generated by the algorithms in the dynamic environment from one of the trials.

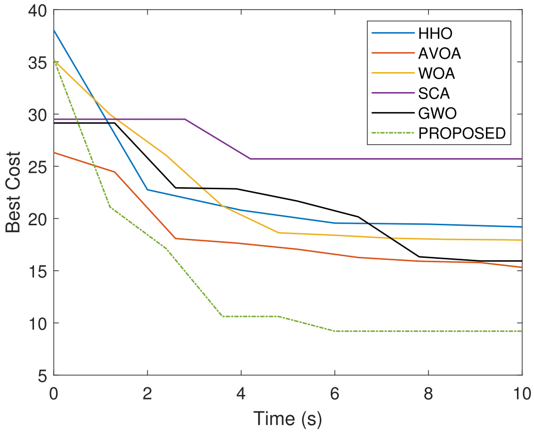

FIGURE 9. Convergence curves corresponding to the paths shown in Fig. 8. It is evident that the proposed method achieves a significantly faster convergence with the lowest best cost compared to the other methods.

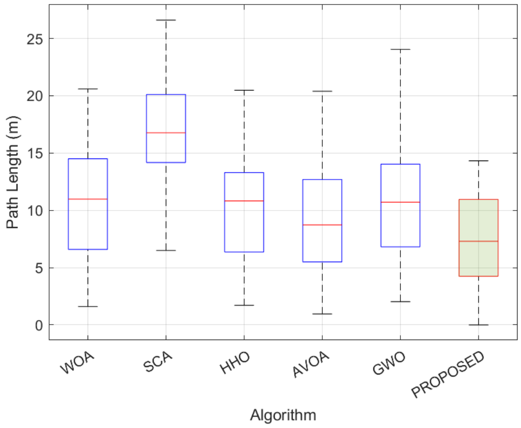

FIGURE 10. Box-plot depicting the comparison of path lengths obtained through the proposed hybrid HHO-AVOA and other algorithms across 50 trials in the dynamic environment.

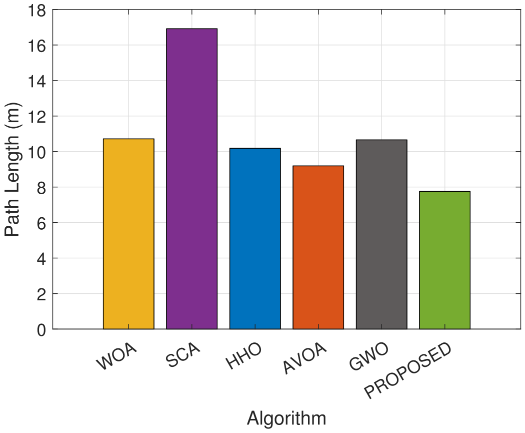

FIGURE 11. Comparison of average path lengths obtained through the proposed hybrid HHO-AVOA against other algorithms based on 50 trials in the dynamic environment.

reach the steady-state value. This consistent trend in fast convergence and lower cost implies that the hybrid HHO-AVOA algorithm retains its low computational complexity even in dynamically changing environments. Moreover, the ability to adapt swiftly and efficiently to alterations in the environment positions the proposed algorithm as a promising solution for real-time applications, particularly in dynamic scenarios where quick decision-making is crucial.

Results from the Monte Carlo analysis for the dynamic environment are showcased in Fig. 10, while the average path length and standard deviation are presented in Figs.11 and 12 respectively. Comparing with the plots from the static case in Fig.4 to Fig.6, a consistent trend can be clearly observed. The proposed algorithm demonstrates superiority by leading in the lowest median value as well as in average path length and standard deviation. Additionally, it reveals the smallest variability as can be seen from the whiskers' length in the box plot. These findings indicate that the performance improvement offered by the proposed algorithm seamlessly extends into dynamic environments.

Fig.13 illustrates the Monte Carlo analysis on the time taken for the DWMR to traverse the paths generated

TABLE 3. Performance improvement based on the average path length and time taken for the DWMR to complete the path for each algorithm in each environment.

		Algorithm						
	Peformance metric	HHO	AVOA	WOA	SCA	GWO	Proposed	Improvement (%)
Static Environment	Path Length (m)	10.79	8.92	9.8	15.92	10.67	7.67	14.01
	Travel duration (s)	19.46	20.31	18.81	3.83	21.95	16.85	17.04
Dynamic Environment	Path Length (m)	10.18	9.19	10.71	16.91	10.65	7.75	27.64
	Travel duration (s)	25.18	22.68	23.46	26.85	30.12	17.08	27.20

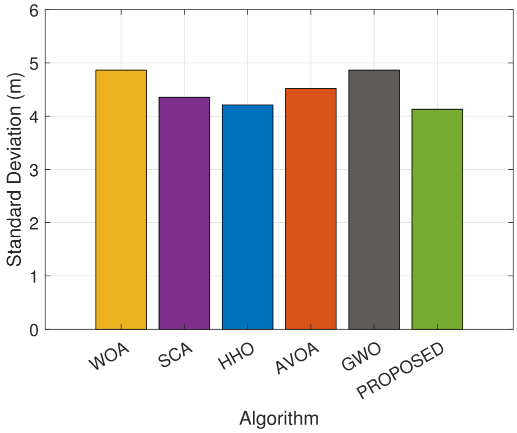

FIGURE 12. Comparison of standard deviations obtained through the proposed hybrid HHO-AVOA against other algorithms based on 50 trials in the dynamic environment.

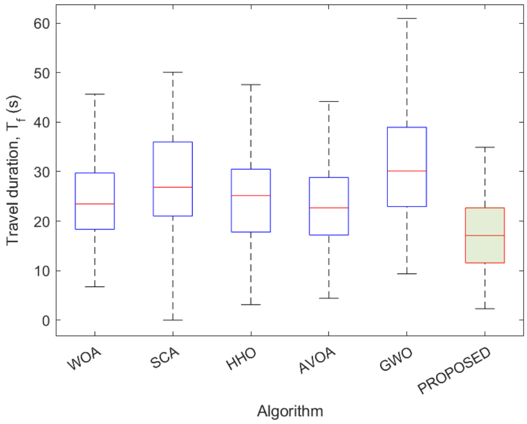

FIGURE 13. Box-plot depicting the comparison of travel durations obtained through the proposed hybrid HHO-AVOA and other algorithms across 50 trials in the dynamic environment.

by each algorithm in the dynamic environment. Similar to the observations in Fig. 10, the proposed method consistently exhibits superior performance across various metrics, including median value and variability.

TABLE 4. The p-value of the Wilcoxon rank-sum test with 5% significance level for both environments.

	Algorithm				
Environment	HHO	AVOA	WOA	SCA	GWO
Static	0.00016	0.00016	0.00010	0.00008	0.00634
Dynamic	0.00016	0.00016	0.00010	0.00008	0.00006

Table 3 provides a quantitative performance comparison between the proposed hybrid HHO-AVOA and other algorithms in both static and dynamic environments in terms of average path length and travel duration. The last column quantifies the performance improvement achieved by the proposed algorithm in comparison to its strongest competitor for each case, which is the AVOA algorithm. As can be observed, there is a 14% reduction in path length and a 17% reduction in travel duration for the static case. Notably, a substantial improvement is observed in the dynamic case, where both path length and travel duration are reduced by 27.64% and 27.2% respectively. These enhancements are attributed to the hybridization of the HHO algorithm with AVOA which contributes to an improved exploitation strategy.

The results of the Wilcoxon rank-sum statistical test which is conducted with a 5% significant level are documented in Table 4. The table indicates that the computed p-value is less than 0.05 which suggests that the proposed HHO-AVOA algorithm effectively balances exploration and exploitation in both static and dynamic environments. The p-value remains consistent for both static and dynamic cases across almost all algorithms as the relative positions of data points within each group are very similar, despite potential differences in raw values. Moreover, the data distribution closely resembles each other in both conditions, resulting in closely aligned ranks and similar values.

IV. CONCLUSION

In this study, we have presented a novel hybrid HHO-AVOA optimization method that is tailored for path planning of a DWMR navigating through both static and dynamic environments while adhering to kinematic constraints. The simulation results and Monte Carlo analysis highlight the substantial improvement provided by the proposed hybrid algorithm when compared to its individual components and other state-of-the-art algorithms. In static scenarios, the hybrid method shows an average 14% reduction in path length and a 17% decrease in DWMR travel duration, while in dynamic scenarios, it demonstrates significant superiority with an average 27.6% reduction in path length and a 27.2% decrease in travel duration. In addition, the proposed algorithm's low computational complexity makes it suitable for real-time implementations, especially in rapidly changing environments that require quick decision-making. In summary, this study establishes a robust foundation for ongoing advancements in DWMR path planning, paving the way for innovative solutions across various industries.

Future works include applications of the hybrid approach to real-world robotic systems which entails exploration to validate its practicality and adaptability. Conducting comprehensive experiments will be paramount to assess the algorithm's performance and feasibility in practical applications. For instance, deploying the hybrid approach in an industrial setting with autonomous robots tasked with navigating complex environments will offer valuable insights into its real-world applicability. This empirical validation will involve systematically testing the algorithm under various conditions, such as dynamic and unpredictable scenarios, to gauge its adaptability and robustness. Additionally, the validation process may encompass the incorporation of sensory data for dynamic obstacle detection and avoidance. Conducting experiments in environments with moving obstacles and dynamically changing conditions, such as crowded public spaces or manufacturing floors with human-robot collaboration, will be essential to verify the algorithm's efficacy in handling real-time obstacles.

CONFLICT OF INTEREST

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

DATA AVAILABILITY

The data supporting the findings of this study are available on request from the corresponding author.

REFERENCES

[1] A. Brunete, A. Ranganath, S. Segovia, J. P. de Frutos, M. Hernando, and E. Gambao, "Current trends in reconfigurable modular robots design," *Int. J. Adv. Robotic Syst.*, vol. 14, no. 3, pp. 2–21, May 2017.

[2] J. H. Teo, A. Loganathan, P. Goh, and N. S. Ahmad, "Autonomous mobile robot navigation via RFID signal strength sensing," *Int. J. Mech. Eng. Robot. Res.*, vol. 9, no. 8, 2020.

[3] Y. Shen, D. Guo, F. Long, L. A. Mateos, H. Ding, Z. Xiu, R. B. Hellman, A. King, S. Chen, C. Zhang, and H. Tan, "Robots under COVID-19 pandemic: A comprehensive survey," *IEEE Access*, vol. 9, pp. 1590–1615, 2021.

[4] I. Arrouch, N. S. Ahmad, P. Goh, and J. Mohamad-Saleh, "Close proximity time-to-collision prediction for autonomous robot navigation: An exponential GPR approach," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 11171–11183, Dec. 2022.

[5] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar, and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*, vol. 21, no. 23, p. 7898, Nov. 2021.

[6] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006.

[7] J. Wang, B. Li, and M. Q.-H. Meng, "Kinematic constrained bi-directional RRT with efficient branch pruning for robot path planning," *Exp. Syst. Appl.*, vol. 170, May 2021, Art. no. 114541.

[8] J. Scharff Willners, D. Gonzalez-Adell, J. D. Hernández, È. Pairet, and Y. Petillot, "Online 3-Dimensional path planning with kinematic constraints in unknown environments using hybrid A* with tree pruning," *Sensors*, vol. 21, no. 4, p. 1152, Feb. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/4/1152

[9] J. Yang, X. Wang, and P. Bauer, "Extended PSO based collaborative searching for robotic swarms with practical constraints," *IEEE Access*, vol. 7, pp. 76328–76341, 2019.

[10] N. Syazreen Ahmad, J. Hui Teo, and P. Goh, "Gaussian process for a single-channel EEG decoder with inconspicuous stimuli and eyeblinks," *Comput., Mater. Continua*, vol. 73, no. 1, pp. 611–628, 2022.

[11] F. Yan, Y.-S. Liu, and J.-Z. Xiao, "Path planning in complex 3D environments using a probabilistic roadmap method," *Int. J. Autom. Comput.*, vol. 10, no. 6, pp. 525–533, Dec. 2013.

[12] O. A. A. Salama, M. E. H. Eltaib, H. A. Mohamed, and O. Salah, "RCD: Radial cell decomposition algorithm for mobile robot path planning," *IEEE Access*, vol. 9, pp. 149982–149992, 2021.

[13] S. Y. Ng and N. S. Ahmad, "Obstacle avoidance strategy for wheeled mobile robots with a simplified artificial potential field," in *Intelligent Computing*, K. Arai, R. Bhatia, and S. Kapoor, Eds. Cham, Switzerland: Springer, 2019, pp. 1247–1258.

[14] M. S. Das, S. Sanyal, and S. Mandal, "Navigation of multiple robots in formative manner in an unknown environment using artificial potential field based path planning algorithm," *Ain Shams Eng. J.*, vol. 13, no. 5, Sep. 2022, Art. no. 101675.

[15] Y. Chen, J. Liang, Y. Wang, Q. Pan, J. Tan, and J. Mao, "Autonomous mobile robot path planning in unknown dynamic environments using neural dynamics," *Soft Comput.*, vol. 24, no. 18, pp. 13979–13995, Sep. 2020.

[16] M. Samadi Gharajeh and H. B. Jond, "An intelligent approach for autonomous mobile robots path planning based on adaptive neuro-fuzzy inference system," *Ain Shams Eng. J.*, vol. 13, no. 1, Jan. 2022, Art. no. 101491.

[17] X. Dai, S. Long, Z. Zhang, and D. Gong, "Mobile robot path planning based on ant colony algorithm with A* heuristic method," *Frontiers Neurorobotics*, vol. 13, pp. 1–9, Apr. 2019.

[18] L. Xu, B. Song, and M. Cao, "A new approach to optimal smooth path planning of mobile robots with continuous-curvature constraint," *Syst. Sci. Control Eng.*, vol. 9, no. 1, pp. 138–149, Jan. 2021.

[19] K. Akka and F. Khaber, "Mobile robot path planning using an improved ant colony optimization," *Int. J. Adv. Robotic Syst.*, vol. 15, no. 3, pp. 1–7, May 2018.

[20] J. Liu, X. Wei, and H. Huang, "An improved grey wolf optimization algorithm and its application in path planning," *IEEE Access*, vol. 9, pp. 121944–121956, 2021.

[21] L. Liu, L. Li, H. Nian, Y. Lu, H. Zhao, and Y. Chen, "Enhanced grey wolf optimization algorithm for mobile robot path planning," *Electronics*, vol. 12, no. 19, p. 4026, Sep. 2023. [Online]. Available: https://www.mdpi.com/2079-9292/12/19/4026

[22] T.-K. Dao, T.-S. Pan, and J.-S. Pan, "A multi-objective optimal mobile robot path planning based on whale optimization algorithm," in *Proc. IEEE 13th Int. Conf. Signal Process. (ICSP)*, Mar. 2016, pp. 337–342.

[23] Z. Yan, J. Zhang, Z. Yang, and J. Tang, "Two-dimensional optimal path planning for autonomous underwater vehicle using a whale optimization algorithm," *Concurrency Comput., Pract. Exper.*, vol. 33, no. 9, p. e6140, May 2021.

[24] A. Loganathan and N. S. Ahmad, "A systematic review on recent advances in autonomous mobile robot navigation," *Eng. Sci. Technol., Int. J.*, vol. 40, Apr. 2023, Art. no. 101343.

[25] Y. Dai, J. Yu, C. Zhang, B. Zhan, and X. Zheng, "A novel whale optimization algorithm of path planning strategy for mobile robots," *Int. J. Speech Technol.*, vol. 53, no. 9, pp. 10843–10857, May 2023.

[26] M. Zhou, Z. Wang, J. Wang, and Z. Dong, "A hybrid path planning and formation control strategy of multi-robots in a dynamic environment," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 26, no. 3, pp. 342–354, May 2022.

[27] Y. Liu and X. Li, "A hybrid mobile robot path planning scheme based on modified gray wolf optimization and situation assessment," *J. Robot.*, vol. 2022, pp. 1–9, Feb. 2022.

[28] Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved grey wolf optimization algorithm and application," *Sensors*, vol. 22, no. 10, p. 3810, May 2022.

[29] R. Kumar, L. Singh, and R. Tiwari, "Path planning for the autonomous robots using modified grey wolf optimization approach," *J. Intell. Fuzzy Syst.*, vol. 40, no. 5, pp. 9453–9470, Apr. 2021.

[30] N. S. Ahmad, "Modeling and hybrid PSO-WOA-based intelligent PID and state-feedback control for ball and beam systems," *IEEE Access*, vol. 11, pp. 137866–137880, 2023.

[31] Q. Si and C. Li, "Indoor robot path planning using an improved whale optimization algorithm," *Sensors*, vol. 23, no. 8, p. 3988, Apr. 2023.

[32] S. Kumar, D. R. Parhi, A. K. Kashyap, and M. K. Muni, "Static and dynamic path optimization of multiple mobile robot using hybridized fuzzy logic-whale optimization algorithm," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 235, no. 21, pp. 5718–5735, Nov. 2021.

[33] R. Akay and M. Y. Yildirim, "Multi-strategy and self-adaptive differential sine–cosine algorithm for multi-robot path planning," *Exp. Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120849.

[34] A. A. Heidari, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[35] A. Loganathan and N. S. Ahmad, "Robot path planning via Harris hawks optimization: A comparative assessment," in *Proc. Int. Conf. Energy, Power, Environ., Control, Comput. (ICEPECC)*, Mar. 2023, pp. 1–4.

[36] G. Li, D. Zhu, Y. Yu, H. Chen, Q. Liu, and S. Yu, "Robot path planning based on improved Harris hawk optimization algorithm," in *Proc. 5th Int. Conf. Intell. Auto. Syst. (ICoIAS)*, Sep. 2022, pp. 209–214.

[37] L. Huang, Q. Fu, and N. Tong, "An improved Harris hawks optimization algorithm and its application in grid map path planning," *Biomimetics*, vol. 8, no. 5, p. 428, Sep. 2023.

[38] F. H. Ajeil, I. K. Ibraheem, A. T. Azar, and A. J. Humaidi, "Grid-based mobile robot path planning using aging-based ant colony optimization algorithm in static and dynamic environments," *Sensors*, vol. 20, no. 7, p. 1880, Mar. 2020.

[39] F. Kamil and M. Y. Moghrabiah, "Multilayer decision-based fuzzy logic model to navigate mobile robot in unknown dynamic environments," *Fuzzy Inf. Eng.*, vol. 14, no. 1, pp. 51–73, Jan. 2022.

[40] T. Tatehara, A. Nagahama, and T. Wada, "Online maneuver learning and its real-time application to automated driving system for obstacles avoidance," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 4, pp. 3144–3153, Apr. 2023.

[41] M. Jafari, H. Xu, and L. R. G. Carrillo, "Brain emotional learning-based path planning and intelligent control co-design for unmanned aerial vehicle in presence of system uncertainties and dynamic environment," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Nov. 2018, pp. 1435–1440.

[42] N. S. Ahmad, "Robust H_∞-fuzzy logic control for enhanced tracking performance of a wheeled mobile robot in the presence of uncertain nonlinear perturbations," *Sensors*, vol. 20, no. 13, p. 3673, 2020.

[43] W. Vega-Brown and N. Roy, *Asymptotically Optimal Planning under Piecewise-Analytic Constraints*. Cham, Switzerland: Springer, May 2020, pp. 528–543.

[44] B. Abhishek, S. Ranjit, T. Shankar, G. Eappen, P. Sivasankar, and A. Rajesh, "Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs," *Social Netw. Appl. Sci.*, vol. 2, no. 11, p. 1805, Nov. 2020.

[45] A. Hasankhani, E. B. Ondes, Y. Tang, C. Sultan, and J. Van Zwieten, "Integrated path planning and tracking control of marine current turbine in uncertain ocean environments," in *Proc. Amer. Control Conf. (ACC)*, 2022, pp. 3106–3113.

[46] N. M. Nakrani and M. M. Joshi, "A human-like decision intelligence for obstacle avoidance in autonomous vehicle parking," *Int. J. Speech Technol.*, vol. 52, no. 4, pp. 3728–3747, Mar. 2022.

[47] Z. Zhang, C. Wang, W. Zhao, and J. Feng, "Longitudinal and lateral collision avoidance control strategy for intelligent vehicles," *Proc. Inst. Mech. Eng., D, J. Automobile Eng.*, vol. 236, nos. 2–3, pp. 268–286, Feb. 2022.

[48] M. S. Abed, O. F. Lutfy, and Q. F. Al-Doori, "Online path planning of mobile robots based on African vultures optimization algorithm in unknown environments," *J. Européen des Syst. Automatisés*, vol. 55, no. 3, pp. 405–412, Jun. 2022.

[49] B. Abdollahzadeh, F. S. Gharehchopogh, and S. Mirjalili, "African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems," *Comput. Ind. Eng.*, vol. 158, Aug. 2021, Art. no. 107408.

[50] A. Pandey and D. R. Parhi, "Optimum path planning of mobile robot in unknown static and dynamic environments using fuzzy-wind driven optimization algorithm," *Defence Technol.*, vol. 13, no. 1, pp. 47–58, Feb. 2017.

[51] S. Y. Chan, N. S. Ahmad, and W. Ismail, "Anti-windup compensator for improved tracking performance of differential drive mobile robot," in *Proc. IEEE Int. Syst. Eng. Symp. (ISSE)*, Oct. 2017, pp. 1–5.

ANBALAGAN LOGANATHAN received the B.Eng. and M.Sc. degrees in electronic engineering from Universiti Sains Malaysia, Nibong Tebal, Malaysia, in 2017 and 2020, respectively, where he is currently pursuing the Ph.D. degree with the School of Electrical and Electronics Engineering. His research interests include optimization algorithms, path planning, and mobile robots.

NUR SYAZREEN AHMAD (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering and the Ph.D. degree in control systems from The University of Manchester, U.K. She is currently an Associate Professor with the School of Electrical and Electronic Engineering, University Sains Malaysia (USM), specializing in intelligent control, sensor networks, and mobile robotics. Her research interests include autonomous mobile robots, with a particular focus on sensing, identification, intelligent control, and indoor navigation. She is a member of the IEEE Young Professional and Control System Societies and has served as a reviewer for several high-impact journals within her field.