Unmanned Systems, Vol. 14, No. 1 (2026) 21–42

DOI: 10.1142/S2301385025500785

Unmanned Systems
https://www.worldscientific.com/worldscinet/us

Optimal Path Planning Algorithm of Indoor AGV Enhanced with HHO and Sector-Wide Field DWA

Zhen Zhou*, Yujie Song, Shuo Li

Key Laboratory of Digital Medical Engineering of Hebei Province,
College of Electronic and Information Engineering Hebei University,
Baoding 071000, P. R. China

With an aim to solve the issue of dynamic obstacles and the global optimal path planning for indoor Automated Guided Vehicles (AGVs), this paper addressed an enhanced hybrid path planning method based on the improvement of Harris Hawk Optimization (HHO) and Dynamic Window Approach (DWA). The main contributions are given as follows. On the one hand, compared to the traditional HHO, the performances are improved in the following aspects. First, initializing the Harris Hawk population with a circle-tent chaotic map to enhance search capability. Second, balancing global and local searches with a segmented strategy to avoid local optima; Third, improving population survival with a hard-constraint mutation strategy. On the other hand, DWA is enhanced from the following three perspectives: a sector potential field is adopted for early obstacle avoidance and a smoother AGV trajectories; a dynamic weighting strategy is used to obtain a better escape local optima and enhance safety; a obstacle-free path attraction strategy is proposed to solve the problem of detours and accelerates the convergence speed. Finally, combined the enhanced HHO with DWA, Generalized Harris Hawks Optimization-Dynamic Window Approach (GHHO-DWA) algorithm addresses the issues of nonsmooth global paths and inadequate dynamic obstacle avoidance. Meanwhile, the effectiveness and practicality of GHHO-DWA algorithm are validated through MATLAB simulations and physical testing on the QBot2e platform.

Keywords: Harris hawk optimization algorithm; path planning; dynamic path planning; sector-wide field.

1. Introduction

In recent years, with the rapid development of artificial intelligence, Automated Guided Vehicles (AGVs) have seen widespread application in the industrial and logistics sectors [1, 2]. AGVs operate autonomously without a human driver and use their own navigation system to navigate predetermined paths, and are primarily used for tasks such as material transportation, loading and unloading, and warehouse management. Owing to their efficiency, flexibility, and cost-saving advantages, AGV systems have been extensively adopted in modern factories and large warehouses [3–5]. Despite the numerous advantages of AGV systems, their path planning faces several challenges. Traditional path planning algorithms often encounter path conflicts and suboptimal path optimization in complex dynamic environments, resulting in decreased AGV operational efficiency. Furthermore, existing path planning methods frequently fail to adequately consider the AGV's own radius and dynamic obstacle avoidance requirements, leading to possible collisions with obstacles and increased safety risks [6, 7]. Therefore, achieving efficient path planning and real-time obstacle avoidance in complex dynamic environments has become a crucial research topic for AGVs.

Path planning algorithms are typically divided into global path planning and local path planning based on their search scope and target tasks. In path planning, global path planning is a crucial step in determining the overall optimal path from the starting point to the target point, which often face challenges such as large search spaces, complex environmental constraints, and path optimization, necessitating the use of effective algorithms and techniques to address these issues [8, 9]. Common methods in global path

Received 30 June 2024; Revised 24 September 2024; Accepted 24 September 2024; Published 10 January 2025. This paper was recommended for publication in its revised form by editorial board member, Yulong Ding.
Email Address: **ouen_zh0617@hbu.edu.cn*
*Corresponding author.

planning include classical algorithms like A* and Dijkstra, as well as swarm intelligence optimization algorithms. Swarm intelligence optimization algorithms, which simulate the behavior of natural swarms, have shown unique advantages in the field of path planning [10]. Through the advantages of global search capability, adaptability and parallelism, these algorithms are able to provide efficient and safe path planning solutions in both static and dynamic environments. Therefore, swarm intelligence optimization algorithms have been widely applied in path planning problems, offering reliable support for system path planning tasks [11, 12].

The Harris Hawk Optimization (HHO) algorithm is widely used in AGV path planning due to its strong global search capability, fast convergence speed, and robust performance. However, the traditional HHO algorithm has certain drawbacks, such as the problem of local optima, sensitivity to parameter selection, lack of safety, excessively long paths, and limited applicability. Many researchers have proposed improvements to address these issues. He *et al.* [13] used segmented chaotic maps to initialize the population and introduced nonlinear dynamic weighting factors to optimize the producer update equation. They also employed an improved sine-cosine algorithm to optimize the searcher update equation, avoiding local optima and improving convergence accuracy. Cai *et al.* [14] proposed a modified HHO algorithm (MHHO) that enhances path planning performance through strategies like a straight-line path strategy, a local search update strategy, and a nonlinear control strategy. These improvements result in smoother paths with reduced length, stronger global search capabilities to avoid local optima, and faster convergence. Li *et al.* [15] introduced a global robot path planning method based on the HHO algorithm, involving coarse and fine-grained planning steps. They first used the MAKLINK graph and Dijkstra's algorithm for coarse-grained path planning, and then applied the HHO algorithm for fine-grained optimization. An improved version, IHHO, was proposed with Rank-based dynamic updates and nonlinear perturbation control factors to avoid local optima. Gregorič *et al.* [16] utilized autonomously generated hierarchical environmental maps, with layered abstractions representing the environment from the most detailed to the most abstract levels. Precomputed partial paths at the most detailed level served as graph edges at the highest level, effectively reducing the complexity of path planning and improving efficiency. Rodríguez *et al.* [17] proposed a new steering angle allocation strategy that, by maintaining a constant speed, generates a set of discrete possible directions for each robot, effectively solving the problem of collision-free path allocation of turning angles and minimizing deviations from the shortest path. Existing improvement strategies for the HHO algorithm typically do not consider the size and dynamic model of AGV, fail to meet AGV motion constraints, and do not account for the presence of dynamic obstacles, making it difficult to satisfy path safety and obstacle avoidance requirements [18, 19]. The aforementioned studies have optimized HHO algorithm in terms of efficiency, computational complexity, and energy efficiency. The use of various optimization strategies and the integration of multiple algorithms offer significant reference value for subsequent research, enhancing algorithm performance and improving the efficiency and safety of path planning to some extent. Despite improvements in HHO algorithm's performance, there remains room for further optimization and enhancement. Moreover, current research on the HHO algorithm primarily focuses on algorithm theory and its ability to solve objective functions. Research on applying the HHO algorithm to robot path planning is still relatively limited, and studies on its applicability in addressing dynamic path planning challenges in complex environments are also scarce.

In local path planning, commonly used methods include the Artificial Potential Field (APF) method and Dynamic Window Approach (DWA), which have the unique advantage of sampling the velocity space, simulating the trajectory, and evaluating its safety, efficiency, and consistency with the target direction to select the optimal velocity command. These algorithms, with their real-time capabilities, flexibility, and adaptability, provide efficient and safe path planning solutions in dynamic environments. Consequently, DWA has been widely applied in local path planning problems, offering reliable support for system path planning tasks. The DWA algorithm is extensively used in AGV path planning due to its strong real-time response capability, effective obstacle avoidance, and high adaptability. However, traditional DWA algorithms also have some disadvantages, such as the problem of local optima, nonsmooth paths, high computational complexity, and insufficient adaptability to dynamic environments. Many researchers have proposed improvements to address these issues. Wu *et al.* [20] combined a stability fuzzy controller with a collision risk controller and defined two proportional factors to improve fuzzy control based on global and obstacle avoidance phases. This significantly reduced the rotation angle of the DWA, shortening the navigation distance and time. Lin *et al.* [21] enhanced the DWA by integrating a fuzzy control strategy to improve obstacle avoidance performance. By introducing collision risk indices and relative distances to assess the threat level of moving obstacles, this method allows for more precise threat evaluation and helps the robot respond quickly and adaptively in complex environments. Zhou *et al.* [22] improved the DWA by adjusting its cost function and introducing additional terms to enhance path tracking and real-time obstacle avoidance for mobile robots. The improved DWA algorithm is more effective in handling dynamic obstacles and generating optimized paths in complex environments. This adjustment of the cost function allows for more flexible obstacle avoidance and better adherence to kinematic constraints. Ding *et al.* [23] used the beetle antennae search algorithm to address the optimization problem of APF, which does not require velocity kinematics, by incorporating randomness in the antenna direction vector. They proposed a collision-free path planning method for cable-driven continuum robots based on an improved APF, solving the problems of local optima and collisions. Although these methods can address high computational complexity and insufficient adaptability to dynamic environments, they do not guarantee a globally optimal path. Existing DWA algorithms mainly have two issues. On the one hand, the DWA is prone to getting trapped in local optima, leading to situations where the target is unreachable. On the other hand, the DWA lacks global path guidance, making it unable to find the optimal path.

As is known to all, single global path planning algorithms struggle to cope with potential unknown static and dynamic obstacles during actual planning processes. Conversely, local path planning algorithms can effectively handle obstacles encountered during travel. But they tend to fall into local optimality, which prevent them from finding a global optimal path [24–26]. Motivated by these observations, the purpose of this paper is to design a new path planning algorithm called the Generalized Harris Hawks Optimization-Dynamic Window Approach (GHHO-DWA), which ensures both global optimality and dynamic obstacle avoidance considering the radius and dynamic performance of AGVs. The main contributions of this paper are as follows:

- First, HHO algorithm is initialized using a circle-tent chaotic map, generating an initial population that is relatively uniform and stable. Then, a segmented global–local search optimization strategy is applied to differentiate between global and local hawks. This approach maintains convergence speed while effectively addressing the problem of the algorithm easily getting trapped in local optima. Furthermore, a hard constraint-mutation strategy is introduced when applying the algorithm to path planning. It enhances the population utilization rate significantly, which is often low due to the randomness of the algorithm during convergence. Consequently, the algorithm can maximize the discovery of the global optimal path with a limited population, and it also significantly improves the efficiency of finding the optimal path. Experimental results in a variety of scenarios demonstrate that the global path planned by GHHO outperforms the global paths generated by three other comparison algorithms in terms of path length, number of turns, the frequency of finding the optimal path, and algorithm runtime.
- The sector potential field obstacle avoidance algorithm significantly improves the anticipation of obstacle avoidance and path safety by assigning artificial potential field repulsion to obstacles within a sector region defined by the current position of the AGV, its distance to the target point, and its orientation change. The dynamic weighting strategy adjusts the orientation weight of the evaluation function to prevent the AGV from falling into local optima. The obstacle-free path attraction strategy solves the problem of detours and accelerates the convergence speed by applying a larger attraction force on the obstacle-free path. Experiments show that the improved DWA algorithm is able to generate smoother and safer paths by jumping out of the local optimum and avoiding obstacles in advance.
- Furthermore, the GHHO-DWA algorithm is proposed through using the global path critical points generated by GHHO as local subgoal points of DWA. So that the combination of the two optimizes the overall performance of the path planning for indoor AGVs, and the effectiveness and practicality of the proposed GHHO-DWA algorithm is verified in different complexity scenarios via Matlab simulations and QBot2e platform experiments.

As shown in Fig. 1, the structure of this paper is given as follows. Section 2 introduces the GHHO algorithm. Section 3

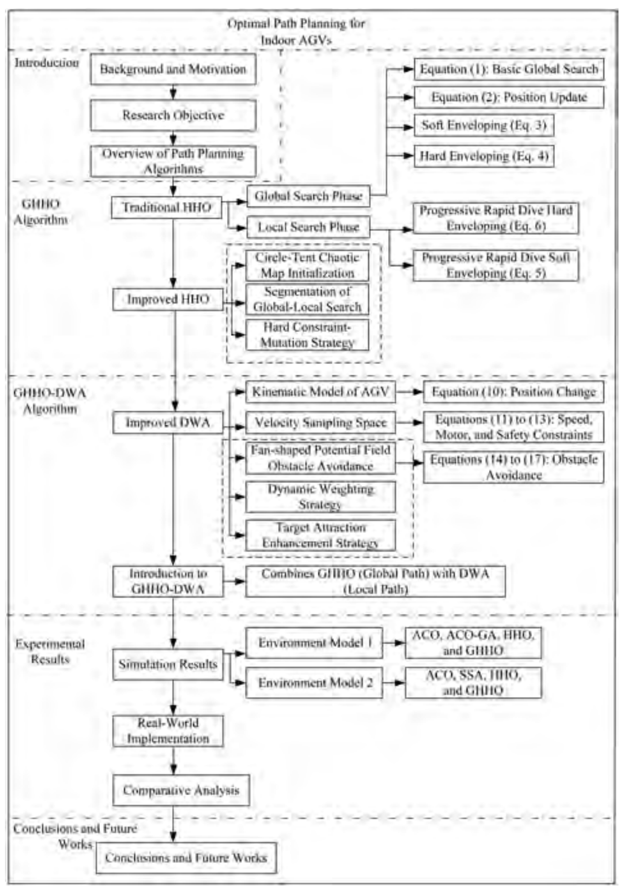


Fig. 1. Overall framework diagram.

presents the improved DWA algorithm and GHHO-DWA algorithm. Section 4 analyzes the simulation and experimental results, which verify the effectiveness and practicality of GHHO-DWA. Section 5 concludes the paper and discusses our future work.

2. Global Path Planning

2.1. *Traditional HHO*

HHO is derived from the hunting behavior of Harris' hawks. When the prey energy factor $|E| \geq 1$, the algorithm performs global search behavior; when $|E| < 1$, it executes local search behavior. Additionally, by introducing the probability of prey successfully escaping the local search behavior is divided into four strategies to simulate the population's predation behavior [27].

2.1.1. *Global search phase*

When $|E| \geq 1$, the algorithm performs the search behavior based on two strategies as follows:

$$x(t+1) = \begin{cases} x_{\text{rand}}(t) - r|x_{\text{rand}}(t) - 2rx(t)| & q \geq 0.5, \\ (x_{\text{rabbit}}(t) - x_{\text{m}}(t)) & \\ \quad -\ r(Lb + rUb - Lb) & q < 0.5, \end{cases} \tag{1}$$

$$x_m(t) = \frac{1}{N}\sum_{i=1}^{N} x_i(t), \tag{2}$$

where t represents the iteration count of the population; $x(t)$ denotes the position of the Harris hawk population; $x_{\text{rabbit}}(t)$ represents the position of the prey; $x_m(t)$ indicates the average position of the individual Harris hawks; r and q are random numbers within $[0,1]$; Ub and Lb are the upper and lower bounds of the algorithm's search space.

2.1.2. *Local search phase*

When $|E| < 1$, the algorithm enters the local search phase. At this point, the algorithm selects four different strategies for predation behavior based on the probability of successful prey escape λ and the value of the prey energy factor $|E|$.

(1) **Soft enveloping:** When $1 > |E| \geq 0.5$ and $\lambda \geq 0.5$, the position update formula for the Harris hawk population during the prey predation process is shown in Eq. (3).

$$x(t+1) = x_{\text{rabbit}}(t) - x(t) - E|Jx_{\text{rabbit}}(t) - x(t)|. \tag{3}$$

(2) **Hard enveloping:** When $|E| < 0.5$ and $\lambda \geq 0.5$, the position update formula for the Harris hawk population during the prey predation process is shown in Eq. (4).

$$x(t+1) = x_{\text{rabbit}}(t) - E|x_{\text{rabbit}} - x(t)|. \tag{4}$$

(3) **Progressive rapid dive soft enveloping:** When $1 > |E| \geq 0.5$ and $\lambda < 0.5$, the position update formula for the Harris hawk population during the prey predation process is shown in Eqs. (5)–(7).

$$Y = x_{\text{rabbit}}(t) - E|Jx_{\text{rabbit}}(t) - x(t)|, \tag{5}$$

$$Z = Y + S \times LF(2), \tag{6}$$

$$x(t+1) = \begin{cases} Y & \text{if } F(Y) < F(x(t)), \\ Z & \text{if } F(Z) < F(x(t)), \end{cases} \tag{7}$$

where $F(\cdot)$ is the fitness function, S is a two-dimensional random vector, and $LF(\cdot)$ is the mathematical expression of Lévy flight.

(4) **Progressive rapid dive hard enveloping:** When $|E| < 0.5$ and $\lambda < 0.5$, the position update formula for the Harris hawk population during the prey predation process is shown in Eqs. (8)–(10).

$$x(t+1) = \begin{cases} Y & \text{if } F(Y) < F(x(t)), \\ Z & \text{if } F(Z) < F(x(t)), \end{cases} \tag{8}$$

$$Y = x_{\text{rabbit}}(t) - E|Jx_{\text{rabbit}}(t) - x_m(t)|, \tag{9}$$

$$Z = Y + S \times LF(2). \tag{10}$$

2.2. *Improvement HHO*

2.2.1. *Circle-Tent chaotic map population initialization*

In the initialization phase, the random generation of the dimensions of the Harris hawk population leads to an uneven distribution of initial solutions, a lack of diversity among individuals, and low coverage of the search space [18]. It may result in premature convergence, slow convergence speed, and insufficient exploration of the solution space. Aimed to address these issues and make ensure a more uniform distribution, the Circle-Tent chaotic map method is applied to initialize the population in this paper.

Chaotic maps are effective tools for solving the problem of uneven population distribution in search algorithm initialization due to their high randomness, chaotic behavior, high-dimensional space coverage, and ease of implementation. Common chaotic mapping strategies include Circle, Tent, Logistic, and Iterative maps. The Circle chaotic map, with its simple structure, can produce chaotic dynamics with periodic control and continuity, demonstrating high coverage in the search space. The formula for the Circle chaotic map is defined in Eq. (11):

$$x_{i+1} = \bmod\left(x_i + 0.2 - \left(\frac{0.5}{2\pi}\sin 2\pi x_i\right)\right), \tag{11}$$

where x_i is the i-th chaotic particle, and x_{i+1} is the $(i+1)$-th chaotic particle.

The Tent chaotic map is a piece wise linear map known for its uniform distribution function and good correlation properties. It is widely used to address the issue of uneven population distribution in search algorithm initialization. The formula for the Tent chaotic map is defined in Eq. (12):

$$x_{i+1} = \begin{cases} x_i/a & x_i \in [0, a), \\ (1 - x_i)/(1 - a) & x_i \in [a, 1], \end{cases} \tag{12}$$

where x_i is the i-th chaotic particle, x_{i+1} is the $(i+1)$-th chaotic particle, and the value of a is 0.499.

The initial population of Harris Hawks Search algorithm is solved based on a Circle-Tent hybrid chaotic map. The formula is defined as follows:

$$x_{i+1} = \begin{cases} x_t \cdot \dfrac{x_i}{a} + x_c \cdot \mathrm{mod}\left(x_i + 0.2 - \left(\dfrac{0.5}{2\pi}\right)\sin(2\pi x_i)\right) \\ \quad x_i \in [0, a), \\ x_t \cdot \dfrac{1 - x_i}{1 - a} + x_c \cdot \mathrm{mod}\left(x_i + 0.2 - \left(\dfrac{0.5}{2\pi}\right)\sin(2\pi x_i)\right) \\ \quad x_i \in [a, 1], \end{cases} \tag{13}$$

where x_i is the i-th chaotic particle, x_{i+1} is the $(i+1)$-th chaotic particle, the value of a is 0.499, the value of x_t is 0.1, and the value of x_c is 0.9.

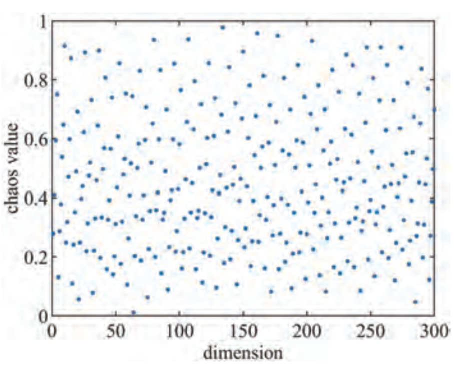
As shown in Fig. 2, particles generated by Circle-Tent hybrid chaotic mapping exhibit a more uniform distribution compared to randomly generated particles, with minimal clustering phenomenon. It indicates the clear chaotic characteristics of Circle-Tent hybrid chaotic mapping, which enhances the complexity and randomness of the chaotic system, aiding in better coverage of the search space.

Remark 1. Circle-Tent hybrid chaotic map combines complex chaotic dynamics of the Circle chaotic system with the Tent chaotic system's fast iteration speed, strong autocorrelation, and suitability for large datasets. In the initialization process of the HHO algorithm, the Circle-Tent hybrid chaotic map is used to generate positions for each dimension of the population, allowing it to cover every part of the search space. This approach enhances global search capability, helps in better discovering the global optimum, improves convergence speed, and thoroughly explores solution space.

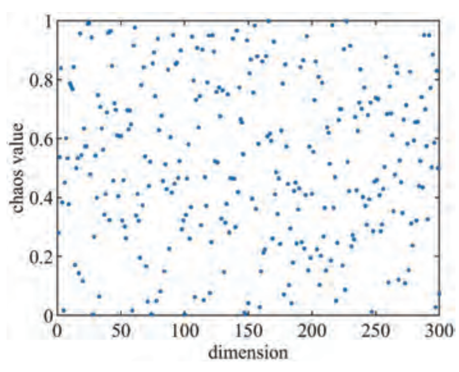
2.2.2. *Segmentation of global-local search*

The prey energy factor E controls the transition of HHO algorithm from global search to local search. As the algorithm iterates, the maximum value of E gradually decreases. When the number of iterations reaches $t \geq T/2$, and $|E| \leq 1$, it indicates that all of Harris' hawks have entered the local search phase. The advantage of this strategy is the gradual narrowing of the search range of the Harris hawk population, thereby enhancing the algorithm's local search capability. However, due to the population becoming overly concentrated, the algorithm often loses diversity in the later stages of iterations. The limited search range here makes the algorithm prone to local optima, thus reducing its search accuracy. To address this issue, this paper proposes a segmentation of global search–local search optimization strategy to prevent the algorithm from falling into local optima.

Based on the principles of swarm intelligence optimization algorithms such as the Bee Algorithm (BA) and Dung Beetle Optimization (DBO) Algorithm, the Harris hawk population is divided into two different subgroups when $t \geq T/2$. One subgroup engages in the global search phase, while the other subgroup performs the local search phase. This approach aims to address the issue of the traditional HHO algorithm being prone to local optima. To retain the local convergence speed of the traditional HHO algorithm, the allocation of the two subgroups becomes a critical parameter under the fixed total number of hawks. Through



(a)



(b)

Fig. 2. (a) Particle plot of Circle-Tent hybrid chaotic mapping (b) Random particle plot.

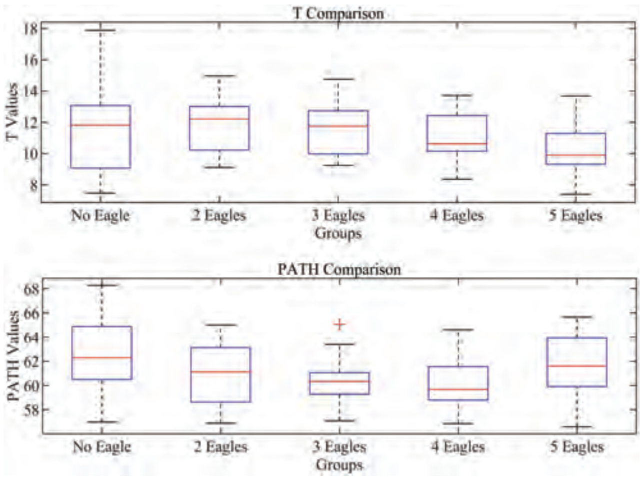


Fig. 3. Comparative illustration of results from 20 runs.

multiple experiments and averaging the results, it was found that allocating four hawks to the global search phase provides the algorithm with a better capability to escape local optima while significantly preserving the local convergence speed. In a 40×40-dimensional map environment, the results after 20 runs are shown in Fig. 3.

Remark 2. By strategically allocating a portion of the hawk population to maintain global search behavior throughout the entire iteration process, the HHO algorithm retains its global exploration capability even in the later stages of the search. This mechanism broadens the exploration range in the early stages and refines solution accuracy through the local hawks' detailed development in the later stages. The dynamic allocation of global and local hawks ensures the algorithm's adaptability at different stages, significantly enhancing overall optimization performance.

2.2.3. *Hard constraint-mutation strategy*

Due to the randomness of the HHO algorithm during the path generation phase in path planning, it cannot consider the positions of obstacles. Instead, paths are evaluated post-generation, and those that fail to avoid obstacles are discarded. Thus, it significantly reduces the efficiency of HHO path planning. Moreover, with a limited number of iterations and a small population size, the probability of finding the optimal solution is greatly reduced. To address this issue, a hard constraint-mutation strategy is introduced into the HHO algorithm during both the global and local search phases. The incorporation of hard constraints and a mutation strategy ensure that each generated HHO position avoids obstacles, thereby significantly improving the survival rate of the population.

Table 1. GHHO algorithm pseudocode.

Description
Initialize parameters: Set population size N, search dimension D, iterations T, and global_hawks_count. Initialize prey escape energy factor E.
Utilize Circle-Tent hybrid chaotic mapping algorithm to initialize the positions X of Harris hawk individuals, generating a matrix of size $N \times D$ to store the position information.
For each Harris hawk i:
Calculate fitness: fitness[i] $\leftarrow$ calculateFitness(x_i).
Record the best fitness value and set the position of the best individual as the rabbit_position.
For $t = 1$ to T:
Calculate prey escape energy E and escape probability.
Based on E:
If $E \geq 1$, perform global exploration.
Else, perform local exploitation.
Check obstacle avoidance:
If an obstacle is encountered, mutatePosition(x_i) to ensure successful obstacle avoidance.
Update fitness:
Calculate new_fitness[i] $\leftarrow$ calculateFitness(x_i).
If new_fitness[i] is better than best_fitness, update best_fitness and rabbit_position.
If $t \geq T/2$:
Divide population into global_group and local_group.
Perform global search on global_group and local search on local_group.
Termination:
If t reaches T, return Best_position and Best_fitness.

Remark 3. For complex constraint optimization problems, this study introduces a hard constraint mutation strategy. When a solution violates hard constraints (where the search mechanism employed by the algorithm cannot account for obstacle positions), the algorithm adjusts the solution through specific mutation operations (mutating dimensions that failed to successfully avoid obstacles), ensuring that it returns to the feasible solution space. This mechanism not only guarantees the feasibility of solutions but also prevents the algorithm from being restricted by constraint conditions, enhancing the adaptability and stability of the HHO algorithm in handling complex constraints.

2.2.4. *Improved algorithm workflow*

The pseudocode representation of the GHHO algorithm is shown in Table 1.

3. Main Results on GHHO-DWA

GHHO addresses and improves upon the limitations of the basic HHO algorithm. Given a start and end point on a two-dimensional grid map, GHHO can generate the globally optimal path from the start to the end point. However, due to the lack of information on dynamic obstacles, GHHO cannot achieve real-time dynamic obstacle avoidance, and it does not consider the size of the AGV itself, resulting in path planning without safety considerations. To address these issues, a fusion algorithm called GHHO-DWA is proposed in this paper, which combines GHHO with an improved DWA algorithm. When the global path is obtained via GHHO, the nodes of the global path are used as target points for DWA, and local motion strategies are formulated based on information about unknown and moving obstacles to achieve global path optimization and local dynamic obstacle avoidance for mobile robots. During the process of local dynamic obstacle avoidance, based on the characteristics of the DWA path planning algorithm, the safety requirements of AGV are achieved.

3.1. *Improvement DWA*

The DWA path planning algorithm generates a series of candidate paths by simulating the possible velocity and angular velocity combinations of an AGV in its current state. These paths are constrained within a dynamic window in the velocity-angular velocity space to ensure that the robot can safely reach the goal within a certain time frame. Subsequently, the quality of each path is evaluated, and the optimal path is selected as the robot's direction of travel. By taking both the dynamic characteristics of the AGV and the dynamic changes in the environment into account, it enables us to navigate quickly and safely in complex environments.

3.1.1. *Kinematic model of AGV*

Assuming that AGV moves forward at a velocity ν and rotates around its center at an angular velocity ω, its position change over a time interval Δt can be described by the following kinematic equations:

$$x_{t+1} = x_t + \nu \cdot \cos(\theta_t) \cdot \Delta t, \tag{14}$$

$$y_{t+1} = y_t + \nu \cdot \sin(\theta_t) \cdot \Delta t, \tag{15}$$

$$\theta_{t+1} = \theta_t + \omega \cdot \Delta t. \tag{16}$$

where (x_t, y_t, θ_t) represents the position and orientation of AGV at time t, and $(x_{t+1}, y_{t+1}, \theta_{t+1})$ represents the position and orientation of the AGV at time $t+1$.

The kinematic model of the AGV is shown in Fig. 4.

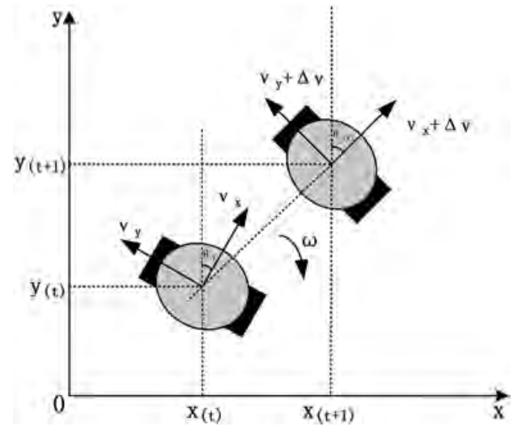


Fig. 4. The kinematic model of an AGV.

3.1.2. *Velocity sampling space*

The velocity sampling space refers to the range of linear and angular velocities that can be sampled within a given time interval [28]. The definition of the velocity sampling space is crucial for determining the appropriate velocity for the AGV's next move.

(1) The speed range of the AGV.

$$\nu_d = \left\{ \begin{matrix} (\nu, \omega) | \nu \in [\nu_c - \dot{\nu}_b \Delta t, \nu_c + \dot{\nu}_a \Delta t] \\ \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t] \end{matrix} \right\}, \tag{17}$$

where $\nu_{\min}$ and $\nu_{\max}$ represent the maximum and minimum linear velocities of the AGV, and $\omega_{\min}$ and $\omega_{\max}$ represent the maximum and minimum angular velocities of the AGV.

(2) The motor constraints of the AGV.
Sampling of velocity and angular velocity is performed within the dynamic window to generate a series of possible velocity and angular velocity combinations. By taking the constraints imposed by the actual motor torque into account, the vehicle is subject to constraints on maximum acceleration and deceleration. The velocity constraint formula is shown in Eq. (18):

$$\nu_m = \{(\nu, \omega) | \nu \in [\nu_{\min}, \nu_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\}, \tag{18}$$

where ν_c and ω_c represent the current linear velocity and angular velocity, respectively, and $\dot{\nu}_a$ and $\dot{\nu}_b$ denote the maximum linear acceleration of the vehicle, while $\dot{\omega}_a$ and $\dot{\omega}_b$ represent the maximum angular acceleration.

(3) The safety constraints of the AGV.
When calculating collision risk, the DWA algorithm introduces the concept of a safety distance, which represents the minimum distance that should be maintained between the vehicle and its obstacles. If the distance between the vehicle and the obstacles is less than

the safety distance, it is considered a collision risk, and corresponding avoidance or deceleration strategies need to be implemented. The safety constraint formula is as shown in Eq. (19).

$$\nu_a = \{(\nu,\omega)|\nu \leq \sqrt{2\text{dist}(\nu,\omega)\dot{\nu}_b},\omega \leq \sqrt{2\text{dist}(\nu,\omega)\dot{\omega}_b}\}, \tag{19}$$

where $\dot{\nu}_b$ and $\dot{\nu}_b$, respectively, represent the maximum linear deceleration and maximum angular deceleration of the vehicle, while $\text{dist}(\nu,\omega)$ denotes the minimum distance between the vehicle and obstacles at the next time step.

3.1.3. *Fan-shaped potential field obstacle avoidance algorithm*

Traditional DWA algorithms can achieve obstacle avoidance for ordinary rectangular obstacles, but they often react when AGV is close to the obstacles, leading to sharp changes in the vehicle's angle and higher risk factors. To address this issue, a Fan-shaped Potential Field Obstacle Avoidance Algorithm is proposed in this paper. The boundaries of a fan-shaped region are defined based on the AGV's current position as the center, the distance from its current position to the target point as the radius, and the angle of the AGV's orientation change at the next time step as the sector angle. Within this fan-shaped region, obstacles are assigned artificial potential field repulsion forces, which are integrated as a new evaluation function to determine the optimal path for the next time step. Experimental results demonstrate that a new evaluation function is introduced to make the vehicle react in advance to obstacles, resulting in safer AGV paths with smoother angle changes.

(1) Calculate the Euclidean distance d_{target} from the current position of the AGV to the target point:

$$d_{\text{target}} = \sqrt{(x_{\text{target}} - x_{\text{AGV}})^2 + (y_{\text{target}} - y_{\text{AGV}})^2}, \tag{20}$$

where $(x_{\text{target}}, y_{\text{target}})$ and $(x_{\text{AGV}}, y_{\text{AGV}})$ represent the target position and the current position of the AGV, respectively.

(2) Change in the orientation of AGV at the next moment:

$$\Delta\theta = \omega_t, \tag{21}$$

$$\theta_{\min} = \theta_t - \frac{\Delta\theta}{2}, \tag{22}$$

$$\theta_{\max} = \theta_t + \frac{\Delta\theta}{2}, \tag{23}$$

where θ_t represents the current orientation of the AGV, and $\Delta\theta$ represents the change in orientation angle.

(3) Calculate the distance d between obstacles within the fan-shaped region and the AGV, as well as the angle a of obstacles relative to the AGV.

$$d = \sqrt{(x_{\text{obs}} - x_{\text{AGV}})^2 + (y_{\text{obs}} - y_{\text{AGV}})^2}, \tag{24}$$

$$a = \arctan 2(y_{\text{obs}} - y_{\text{AGV}}, x_{\text{obs}} - x_{\text{AGV}}) - \theta_{\text{AGV}}, \tag{25}$$

where $(x_{\text{obs}}, y_{\text{obs}})$ represents the position of obstacles within the fan-shaped region.

(4) Define the repulsive force $F_{\text{repulsive}}(d,a)$ represented by a Gaussian function.

$$F_{\text{repulsive}}(d,a) = k \cdot e^{-\frac{(d-d_0)}{2\delta^2}}, \tag{26}$$

where d_0 is the mean of Gaussian function, δ is the standard deviation, and e is the base of the natural logarithm.

(5) Utilizing $F_{\text{repulsive}}(d,a)$ as a new evaluation function, combined with other evaluation functions, to determine the optimal path for the next time step.

Based on the data above, draw the corresponding schematic diagram of the sector potential field obstacle avoidance algorithm, as shown in Fig. 5.

The blue sector area represents the search area assigned with obstacle repulsion, the orange region represents the assigned obstacle repulsion, and the green denotes the target point.

The experimental results of its obstacle avoidance effect in the grid map are shown in Fig. 6.

From Fig. 6, it is evident that in the presence of obstacles ahead, the original algorithm reacts only when it gets very close, resulting in drastic changes in the AGV's angle and low safety. In contrast, the improved algorithm can react in advance under the same conditions, implementing slight turning strategies beforehand, ensuring the AGV reaches

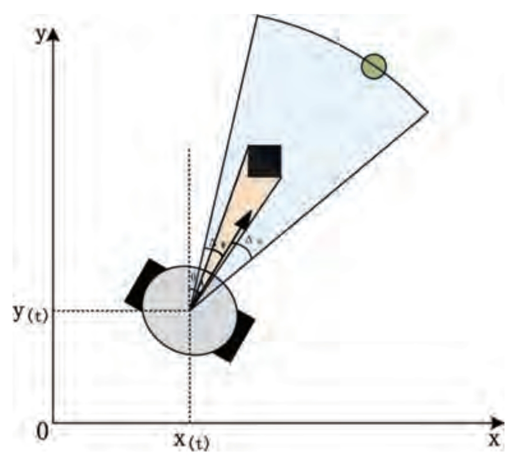
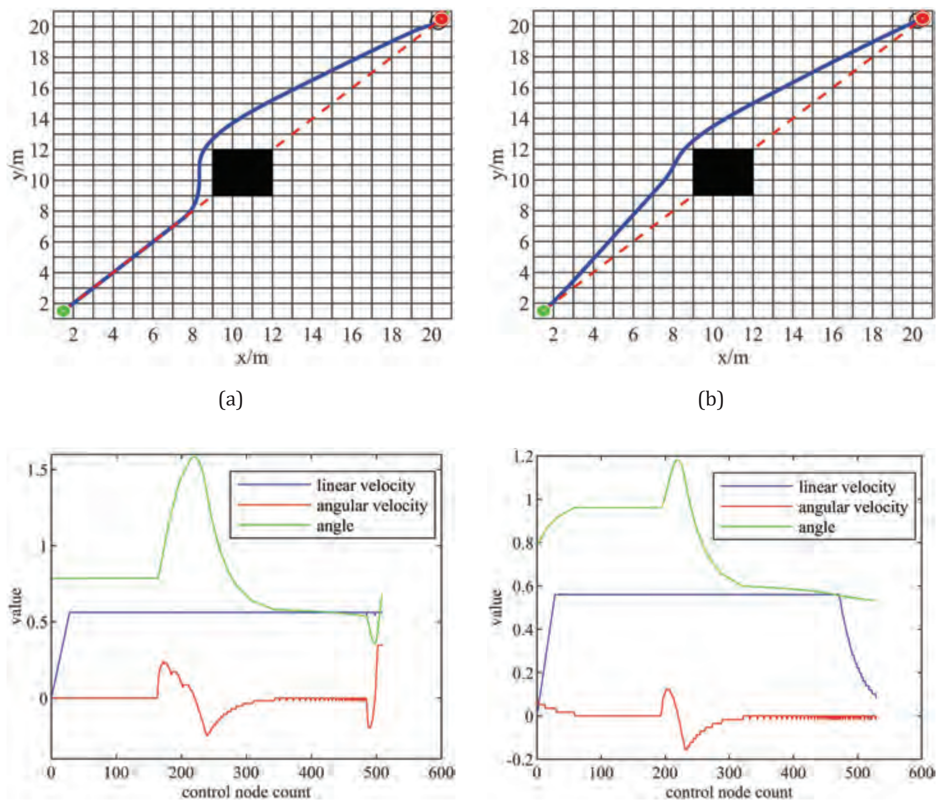


Fig. 5. Schematic diagram of sector potential field obstacle avoidance algorithm.



(d) (d)

Fig. 6. (a) Original DWA algorithm path planning graph (b) Improved DWA algorithm path planning graph (c) Original DWA algorithm linear velocity, angular velocity, angle change graph (d) Improved DWA algorithm linear velocity, angular velocity, angle change graph.

the target point safely without decelerating and with minimal angle changes.

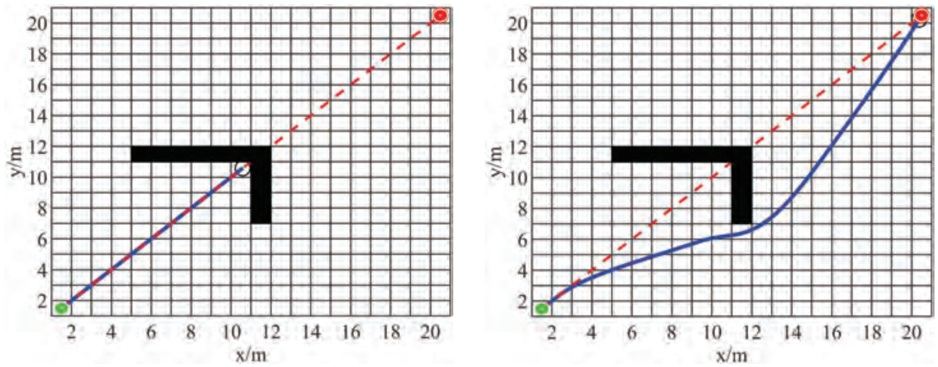
Remark 4. The fan-shaped potential field mechanism defines a fan-shaped region between the AGV's current position and the target point, applying repulsive forces to obstacles within this region. The repulsive force is used as a fourth evaluation function, combined with the other three functions to assess the path cost. The robot plans its path based on the aggregated path costs, prioritizing paths with lower costs to avoid high-density obstacles. This strategy enhances the DWA algorithm's obstacle avoidance capability, allowing the robot to handle complex environments more flexibly and improving overall navigation safety and reliability.

3.1.4. *Dynamic weighting strategy*

Due to the inherent characteristics of AGVs, such as constraints on angular velocity and speed, the original DWA algorithm's three evaluation functions tend to trap AGVs in local optima. To address this issue, this paper proposes a dynamic weighting strategy. It assesses whether there are continuous obstacles in the fan-shaped region that could lead the vehicle into local optima. If such obstacles are detected, the orientation weight of the evaluation function is altered to a negative value; otherwise, the original evaluation function is retained. Experimental results demonstrate that this approach enables the algorithm to escape local optima prematurely.

From Fig. 7, it is evident that the original DWA path planning algorithm gets trapped in local optima when facing concave obstacles, preventing the vehicle from reaching the target point. In contrast, the improved algorithm can identify the type of obstacle ahead and react promptly by turning early to break free from local optima.

Remark 5. This paper proposes a method for dynamically adjusting the evaluation function weights. Specifically, by assessing the presence of continuous obstacles within a fan-shaped region, the strategy dynamically modifies the direction weight. If obstacles that could lead to local optima are detected, the direction weight is adjusted to a negative value, encouraging the algorithm to escape local optima; otherwise, the original weight is maintained. This strategy



(a)

(b)

Fig. 7. Comparison of DWA algorithm path planning diagrams.

significantly enhances the flexibility and success rate of path planning, enabling AGVs to more effectively address challenges in complex environments.

3.1.5. *Target attraction enhancement strategy*

In the above experiments, although the possibility of falling into local optima has been addressed, there is still a concern when turning that, if continuous obstacles persist within the range of the orientation angle, the evaluation function with negative orientation weight will still be used. This situation may result in the occurrence of detours. To address this issue, this paper proposes a new approach to determine whether there are obstacles between the current position and the target point. If there are no obstacles, a new evaluation function is adopted, which can be understood as giving an AGV a stronger attraction towards the target point when there are no obstacles between the two points. Experimental validation not only resolves the aforementioned issue but also accelerates the convergence speed of the algorithm.

Remark 6. To address the potential issue of detours during turning, a new evaluation function is proposed to determine whether there are obstacles between the current position and the target point. If no obstacles are present, the new evaluation function is used to increase the AGV's attraction towards the target point. This approach resolves the detour problem and accelerates the algorithm's convergence speed.

3.2. *Introduction of GHHO-DWA algorithm*

The GHHO algorithm performs remarkably well in acquiring global optimal paths in static obstacle environments with known maps. However, when applied to AGV path planning in real scenarios, the generated global paths lack smoothness and safety considerations. Additionally, they do not account for newly introduced obstacles, rendering the real-time obstacle avoidance ineffective. Conversely, DWA updates an AGV's perception range as a window in real-time to detect nearby unknown and dynamic obstacles. It employs an evaluation function to endow the AGV with dynamic obstacle avoidance capabilities. However, DWA tends to get trapped in local optima when the destination is far, leading to scenarios where the goal cannot be reached or the path is excessively long.

The main process of the GHHO-DWA algorithm is as follows: in a known map environment, GHHO plans the global path and uses the critical nodes of the GHHO-generated global path as the local path target points for DWA. This algorithm employs segmented dynamic path planning to ensure real-time obstacle avoidance capability for the AGV. Moreover, based on the DWA safety distance evaluation function, path safety is ensured. The resulting path is both globally optimal and safe. The GHHO-DWA algorithm combines the advantages of both methods, resulting in shorter paths while enhancing path smoothness and safety. The workflow pseudocode for the GHHO-DWA algorithm is presented in Table 2.

One of the advantages of the GHHO-DWA algorithm lies in its ability to balance the computational costs of global search and local optimization. Although the complexity of the HHO component increases with the population size and the number of iterations, the efficiency is improved through carefully designed initialization and search strategies, which help reduce the generation of invalid search paths. Meanwhile, the DWA component's local optimization, while adding some computational load during each local path adjustment, effectively prevents the algorithm from getting stuck in local optima. Therefore, the complexity of the GHHO-DWA algorithm is not only determined by the computational costs of individual components but is also influenced by the

Table 2. Enhanced AGV path planning using HHO and sector-wide field DWA.

Start
Initialize relevant parameters.
Initialize the hawk population using Circle-Tent Chaotic Mapping.
Calculate and rank individual fitness values:
Update the prey position to the optimal individual position.
Update prey energy factor and escape energy factor.
If $E \geq 1$:
If $q \geq 0.5$:
Perform a global search using Eq. (1-1).
Else:
Perform a global search using Eq. (1-2).
End If
Else:
If $E \geq 0.5$:
Perform a local search using Eq. (3) or (7).
Else:
Perform a local search using Eq. (4) or (8).
End If
If $t \geq T/2$:
Reserve four hawks for global search using Equation (1).
Check if the individual's path successfully avoids obstacles:
If obstacles are present:
Introduce mutation strategy.
Update the population of Harris's Hawks and the global best solution.
If iteration limit or precision condition is reached:
Output the best fitness and the best position of Harris's Hawks.
Use key points of the global path planned by the GHHO algorithm for the DWA algorithm:
The first point of the global path as the starting point.
The next key point as the local subgoal point.
Perform the AGV speed sampling and trajectory prediction:
Introduce the sector potential field obstacle avoidance model using Eq. (26).
If there are dangerous obstacles:
Adjust orientation weight.
Select the optimal trajectory based on the evaluation function.
Move according to the selected trajectory.
Arrive at the local subgoal point:
If the subgoal point is the global goal point:
Generate the optimal path.
End
Output the global optimal path and its corresponding fitness value.
End

interaction and synergy between the global and local search processes. Through this interaction, the algorithm can maintain efficient global search while achieving more refined local optimization, thus demonstrating superior performance in more complex environments.

The GHHO-DWA algorithm combines the strengths of HHO and DWA. The global search capability of HHO ensures that the algorithm explores potential global optima within a broad search space, and its convergence can be demonstrated by proving that it can find the global optimum with high probability. Meanwhile, DWA's local search capability further optimizes the path when approaching the optimal solution. Theoretically, as long as the HHO algorithm gradually converges on a global scale, the DWA algorithm can refine the local path to improve the solution quality. Therefore, the GHHO-DWA algorithm is convergent and, with proper parameter settings, can converge to the global optimum or a near-optimal solution with high probability.

4. Simulation and Experimental

4.1. *Environment modeling*

Environment modeling is a critical step in robotic path planning. It abstracts the real-world environment into a form that can be processed by a computer, enabling the path planning algorithm to understand and operate within it. Common methods for environment modeling in path planning include grid-based methods, visibility graphs, and feature maps. In this paper, a grid-based method is used for binarization, creating a grid map where 0 represents free traversable space and 1 represents obstacles, as shown in Fig. 8. By modeling the actual environment as a grid map, the necessary input for path planning is provided. The path planning algorithm utilizes this grid map to search for the most efficient path from the start point to the destination, adhering to specific constraints and optimization objectives.

$$\mathrm{Map}(i,j)\begin{cases} 0 & \text{Free space,} \\ 1 & \text{Obstacle,} \end{cases} \tag{27}$$

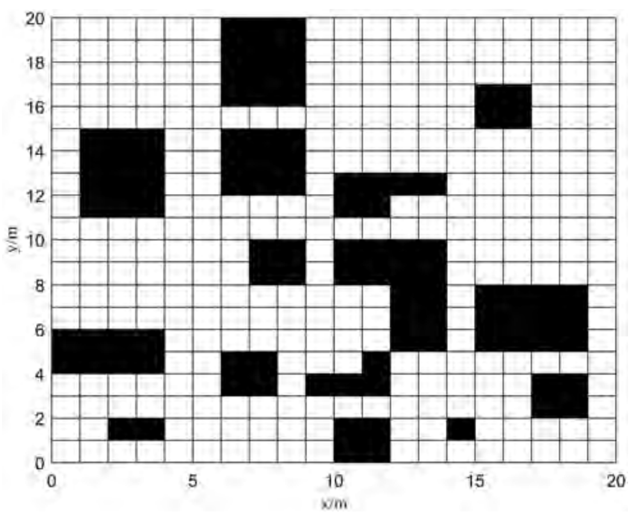


Fig. 8. 2D grid map of a complex environment.

Table 3. Parameter settings for HHO, ACO, ACO_GA, GHHO, and SSA.

HHO	Parameters
Number of Harris Hawks	50
Maximum Number of Iterations	200
Number of Harris's Hawks in GHHO	4
Search Dimension	18
$x_{\mathrm{tent}}/x_{\mathrm{circle}}$	0.1/0.9
ACO	
Ant Quantity	50
Maximum Number of Iterations	200
Pheromone Factor	2
Heuristic Function Factor	6
Pheromone Evaporation Factor	0.1
ACO_GA	
Ant Quantity	50
Maximum Number of Generations	200
Crossover Probability ($p_{\mathrm{crossover}}$)	0.8
Mutation Probability (p_{mutation})	0.05
Pheromone Evaporation Factor	0.1
GHHO	
Number of Harris Hawks	50
Maximum Number of Iterations	200
Number of Harris's Hawks in GHHO	4
Search Dimension	18
$x_{\mathrm{tent}}/x_{\mathrm{circle}}$	0.1/0.9
SSA	
Number of Sparrows	50
Maximum Number of Iterations	200
Soft Value	0.8
Recovery Percentage	0.3
Scout Percentage	0.2

To ensure the reliability and reproducibility of the experimental results, this paper adopts the control variable method. The constant factors in all simulation experiments include the software and hardware configurations. The computer processor used is an Intel(R) Core(TM) i5-1035G1 CPU with a base frequency of 1.00 GHz and an actual operating speed of 1.19 GHz. The operating system is Windows 10 Professional, and the simulation platform is MATLAB R2021b.

4.2. *Simulation on GHHO*

In order to verify the effectiveness and superiority of GHHO, simulations are performed in this chapter on complex grid maps to compare GHHO with the conventional HHO, Ant Colony Optimization (ACO) and Ant Colony Optimization_ Genetic Algorithm (ACO_GA) algorithms. The parameter settings for each algorithm are shown in Table 3.

4.2.1. *Environment model 1*

Environment model 1 was modeled as a 20×20 grid map with randomly distributed obstacles with given start and end points and simulated in MATLAB. Figure 9 intuitively represents the path planning results of ACO [29], ACO_GA [30], HHO, and GHHO. To minimize the influence of randomness in swarm intelligence optimization algorithms, each algorithm is independently run for 30 repetitions of simulation experiments. The comparative results are shown in Fig. 10 and Tables 4 and 5.

The simulation results indicate that all four algorithms can plan collision-free global paths. Analysis was conducted based on the path length, number of turns, and planning time from 30 simulated experiments. The simulation results are shown in Tables 4 and 5. GHHO exhibited outstanding efficiency in path planning. Compared to ACO, HHO, and ACO_GA, the average path length of GHHO was reduced by 11.04%, 3.63%, and 1.32% respectively. Out of the 30 simulated experiments, GHHO, ACO, HHO and ACO_GA found the optimal path 21, 3, 2, and 11 times respectively (as shown in Table 4). The variance of path lengths was 0.0293, 3.0051, 3.0051 and 0.4874 respectively. As depicted in Fig. 9, GHHO demonstrated remarkable stability and the ability to find optimal paths consistently. Regarding the number of turns, GHHO exhibited fewer turns, reducing the number of turns by 36.9%, 14.7% and 8.7% compared to ACO, HHO and ACO_GA respectively. Figure 10 vividly illustrates that GHHO algorithm can reach the endpoint with fewer turns. In terms of timing for path planning, GHHO performed excellently, with an average runtime reduction of 2.3% and 86.2% compared to ACO and ACO_GA, respectively.

4.2.2. *Environment model 2*

To highlight the advantages of the algorithms, Environment Model 2 was modeled as a 40×40 grid map with more complex obstacles added for simulation experiments. Figure 11 visually represents the path planning results of ACO, Sparrow Search Algorithm (SSA) [31], HHO and GHHO. After 30 simulation experiments, the comparison results are shown in Fig. 12 and Tables 6 and 7.

In Environment Model 2, due to the scale and complexity of the map, the shortest path is defined as a path shorter than 57 m. Among the four algorithms' simulation results, GHHO produced the shortest path at 56.4764 meters, reducing the shortest path by 3.74%, 1.76% and 1.04% compared to ACO, HHO and SSA, respectively (as shown in Table 6). In terms of average path length, GHHO reduced the length by 4.48%, 4.33% and 3.84% compared to the other algorithms. There was a significant improvement in the average number of turns, with reductions of 23.7%,

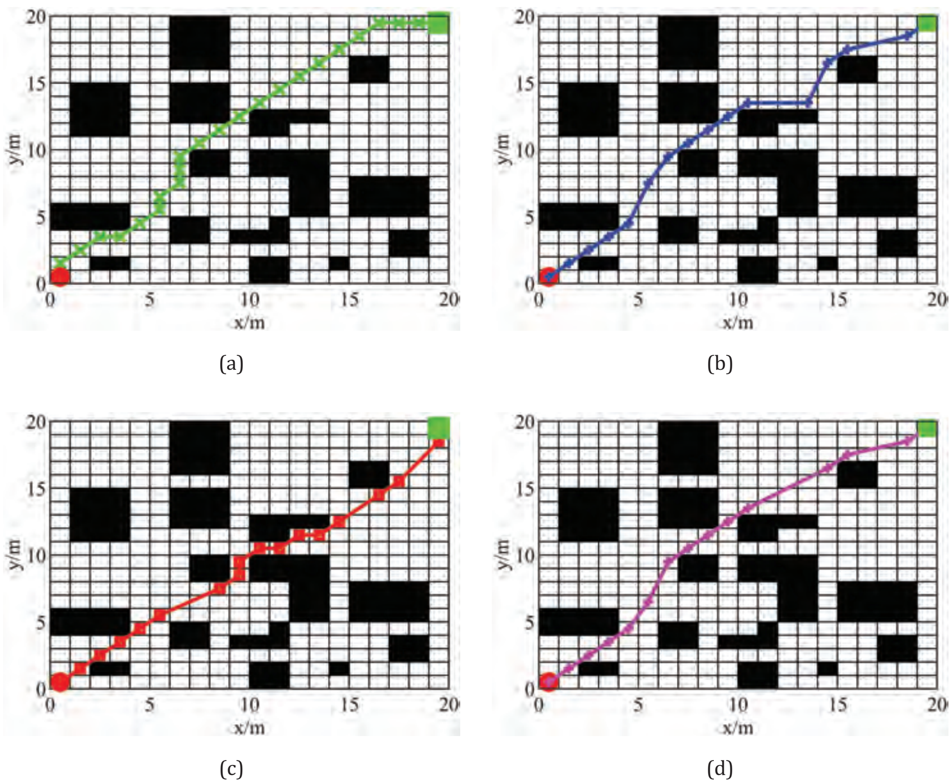


Fig. 9. Trajectories of four path planning algorithms in environment model 1.

10.4% and 15.9% compared to the other algorithms. Tables 6 and 7 shows that in high-scale and high-complexity map environments, GHHO found the optimal path 8 times, while the other three algorithms did not find the optimal path at all. Figure 12 clearly illustrates that the GHHO algorithm can reach the destination with shorter and smoother paths. Furthermore, GHHO demonstrated stronger stability. Although the path planning time of GHHO may be relatively longer as the complexity and scale of the map increase, the resulting stability and shortest path effect are quite impressive.

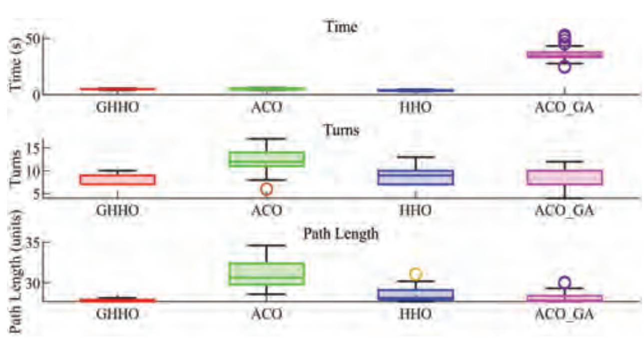


Fig. 10. Performance metrics comparison of algorithms (GHHO, ACO, HHO, ACO_GA) under environment model 1.

The above experimental results indicate that the GHHO proposed in this paper can plan globally optimal paths in maps of varying scales and complexities. The generated global paths consistently outperform the other three comparative algorithms in metrics such as path length, number

Table 4. Comparison of different algorithms (part 1).

Algorithm	Path length/m Minimum	Path length/m Mean	Frequency of minimum value
ACO	29.2132	31.2603	3
HHO	27.7028	28.8564	2
ACO_GA	27.7318	28.1795	11
GHHO	27.7028	27.8083	21

Table 5. Comparison of different algorithms (part 2).

Algorithm	Number of turns Minimum	Number of turns Mean	Time/s Minimum	Time/s Mean
ACO	8	12.2	4.18	5.08
HHO	7	9.03	2.92	3.82
ACO_GA	7	8.43	24.47	36.32
GHHO	7	7.7	4.13	4.97

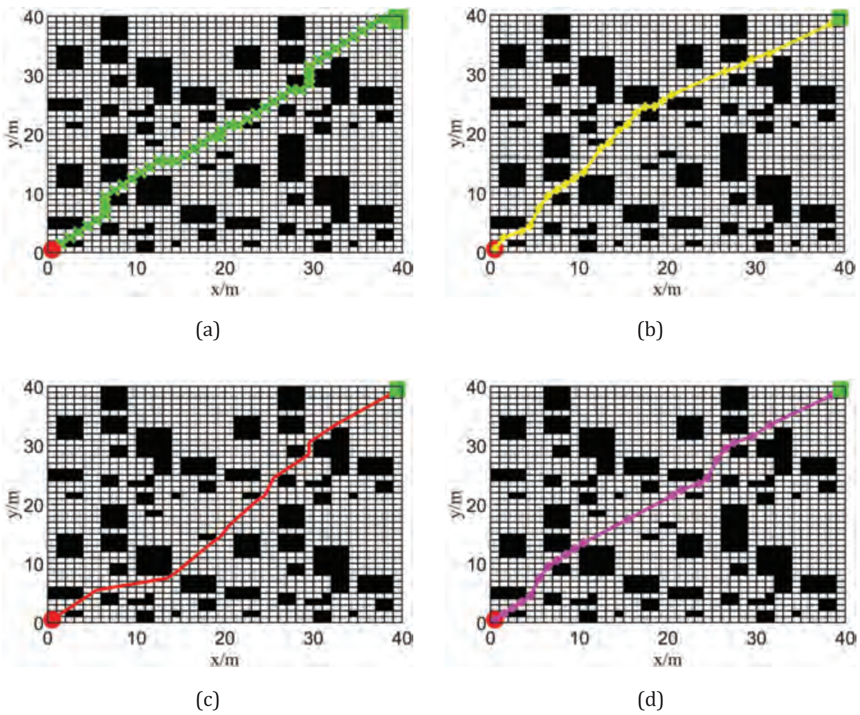


Fig. 11. Trajectories of four path planning algorithms in environment model 2.

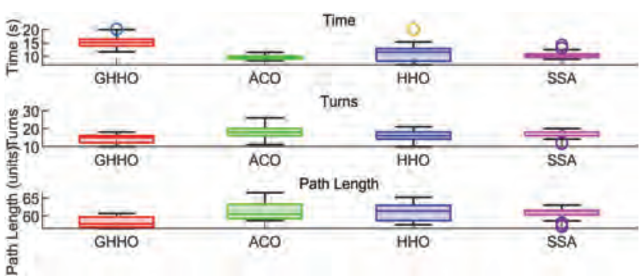


Fig. 12. Performance metrics comparison of algorithms (GHHO, ACO, HHO, SSA) under environment model 2.

of turns, and optimization capability. GHHO not only demonstrates outstanding path optimization capability but also shows the ability to produce smooth paths and excellent stability.

Table 6. Comparison of different algorithms (part 1).

Algorithm	Path length/m		Frequency of minimum value
	Minimum	Mean	
ACO	58.669	61.0529	0
HHO	57.4853	60.9549	0
SSA	57.0709	60.6464	0
GHHO	56.4764	58.3168	8

4.3. *Simulation on GHHO-DWA*

4.3.1. *GHHO-DWA for unknown obstacle avoidance*

Validation of the fused algorithm is conducted on grid maps with three sets of scenarios featuring different configurations of unknown obstacles. The path planning is performed using both GHHO and GHHO-DWA algorithms. The starting and ending points are set to (1, 1) and (50, 50), respectively. Black grids represent known obstacles, while orange grids depict unknown obstacles. The weight parameters for DWA are configured as shown in Table 8, and

Table 7. Comparison of different algorithms (part 2).

Algorithm	Number of turns		Time/s	
	Minimum	Mean	Minimum	Mean
ACO	11	18.7	8.38	9.51
HHO	10	15.93	6.77	11.08
SSA	12	16.97	8.92	10.46
GHHO	10	14.27	11.65	15.17

Table 8. DWA weight parameters.

Parameters	α	β	γ	ε
Weight	0.05	0.1	0.3	0.2

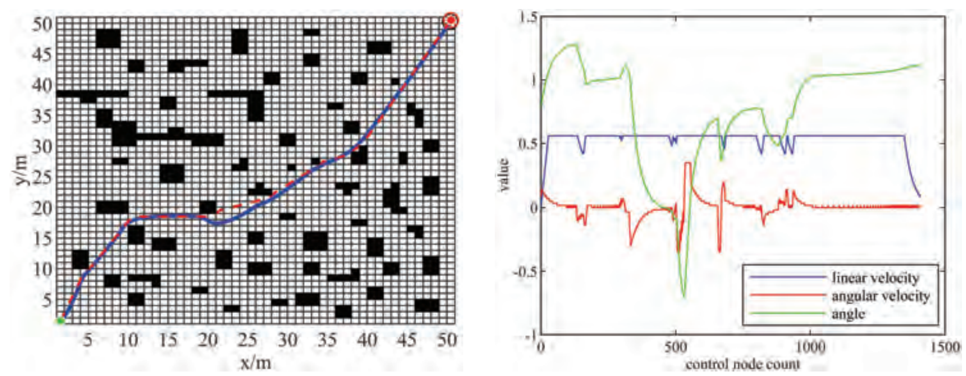
Table 9. AGV kinematic parameters.

Parameter	Numerical values
Starting point coordinates	(1,1)
End point coordinates	(50,50)
Maximum linear velocity of AGV	1.5 m/s
Maximum angular velocity of AGV	20 deg/s
Maximum linear acceleration of AGV	0.2 m/s^2
Maximum angular acceleration of AGV	50 deg/s^2
Linear velocity resolution	0.02 m/s
Angular velocity resolution	1 deg/s
Time resolution	0.1 s
Prediction period	3 s

the dynamic parameters of the AGV are set, as shown in Table 9.

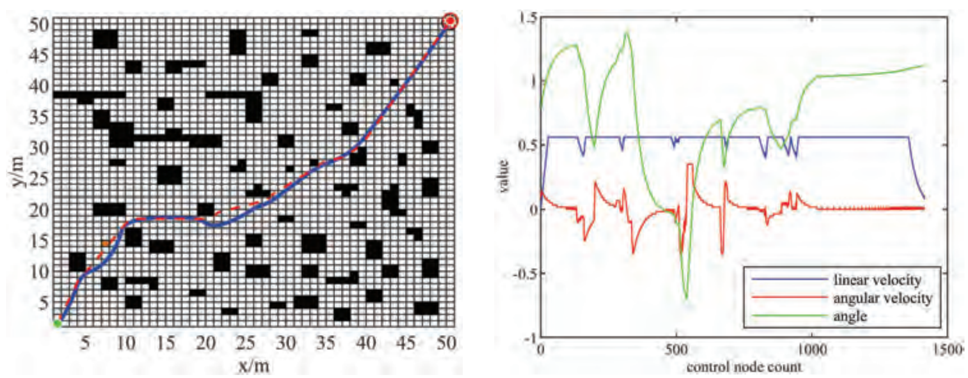
Figures 13–15 depict the simulation results of obstacle avoidance algorithms. The paths planned by GHHO are represented by red dashed lines, while those planned by GHHO-DWA are represented by blue solid lines. Unknown obstacles are denoted by orange rectangles. From these simulation results, it is evident that in the absence of unknown obstacles, both algorithms can achieve optimal paths. However, when encountering unknown obstacles, using GHHO algorithm alone cannot identify and avoid these obstacles. In contrast, GHHO-DWA algorithm, integrated with DWA, can recognize and circumvent unknown obstacles while planning smooth and safe paths. Regardless of the increase in the number of unknown obstacles, the planned paths consistently succeed in obstacle avoidance and reaching the destination. The incorporation of hybrid methods effectively handles unknown obstacles in grid maps, thus providing a trajectory closely associated with the global optimal path. Considering the working characteristics of AGVs in real environments, this simulation experiment uses four evaluation indicators: optimizing path length, number of turns, safety, and smoothness. The relevant performance indicators for random obstacle avoidance using the integrated algorithm are shown in Table 10.

To visually demonstrate the improvement of the GHHO-DWA algorithm, we conducted simulation experiments



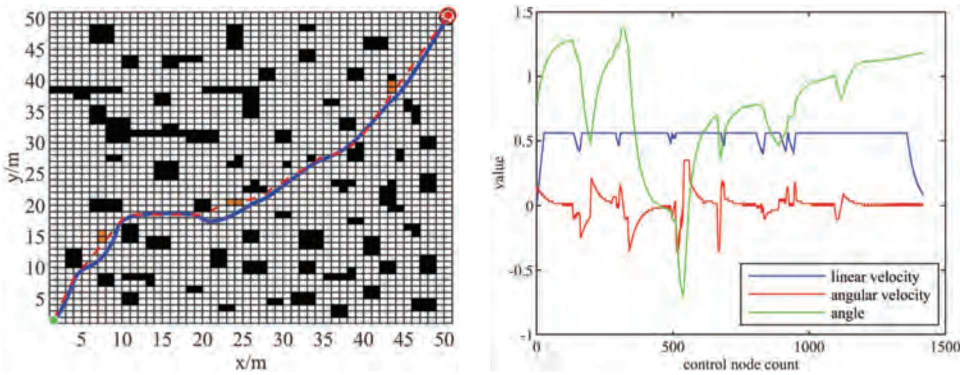
(a) (b)

Fig. 13. No unknown obstacle scenario: (a) Path planning graph, (b) Linear velocity, angular velocity, and angle change graph.



(a) (b)

Fig. 14. A scenario with an unknown obstacle: (a) Path planning graph, (b) Linear velocity, angular velocity, and angle change graph.



(a) (b)

Fig. 15. Multiple unknown obstacle scenario: (a) Path planning graph, (b) Linear velocity, angular velocity, and angle change graph.

Table 10. Path-planning performance of the GHHO-DWA fusion algorithm.

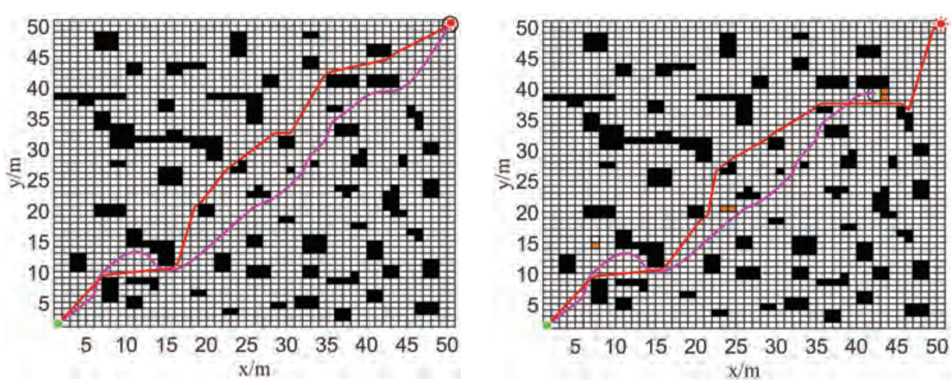
Scenario	Path length	Turns	Smoothness	Safety
Figure 11	75.10	2	Yes	Yes
Figure 12	75.44	3	Yes	Yes
Figure 13	75.52	3	Yes	Yes

under the same conditions for the traditional HHO and DWA algorithms. Performance comparisons were made in both scenarios, without random obstacles and with multiple random obstacles. The specific simulation results are shown in Fig. 16.

Based on the analysis of Tables 10 and 11, the GHHO-DWA fusion algorithm significantly outperforms the traditional HHO and DWA algorithms in path planning. In the absence of random obstacles, GHHO-DWA achieves a path length 0.13 m shorter than HHO, approximately 99.83% of HHO's path length, and 1.34 m shorter than DWA, about 98.24% of DWA's path length. In scenarios with multiple random obstacles, GHHO-DWA's path length is 3.75 m shorter than HHO, about 95.30% of HHO's path length. In terms of the number of turns, GHHO-DWA requires 7 fewer turns than HHO in the absence of random obstacles, representing only 22.22% of HHO's turns, and 6 fewer turns than DWA, about 25.00% of DWA's turns. In scenarios with multiple random obstacles, GHHO-DWA requires 7 fewer turns than HHO, approximately 30.00% of HHO's turns. Additionally, GHHO-DWA achieves path smoothness and safety in all scenarios, whereas HHO and DWA fall short in these aspects. Overall, the GHHO-DWA fusion algorithm demonstrates superior efficiency, smoothness, and safety in path planning, especially in complex environments with random obstacles.

4.3.2. *GHHO-DWA for dynamic obstacle avoidance*

To assess the obstacle avoidance capability of GHHO-DWA fusion algorithm in dynamic environments, simulations



(a) (b)

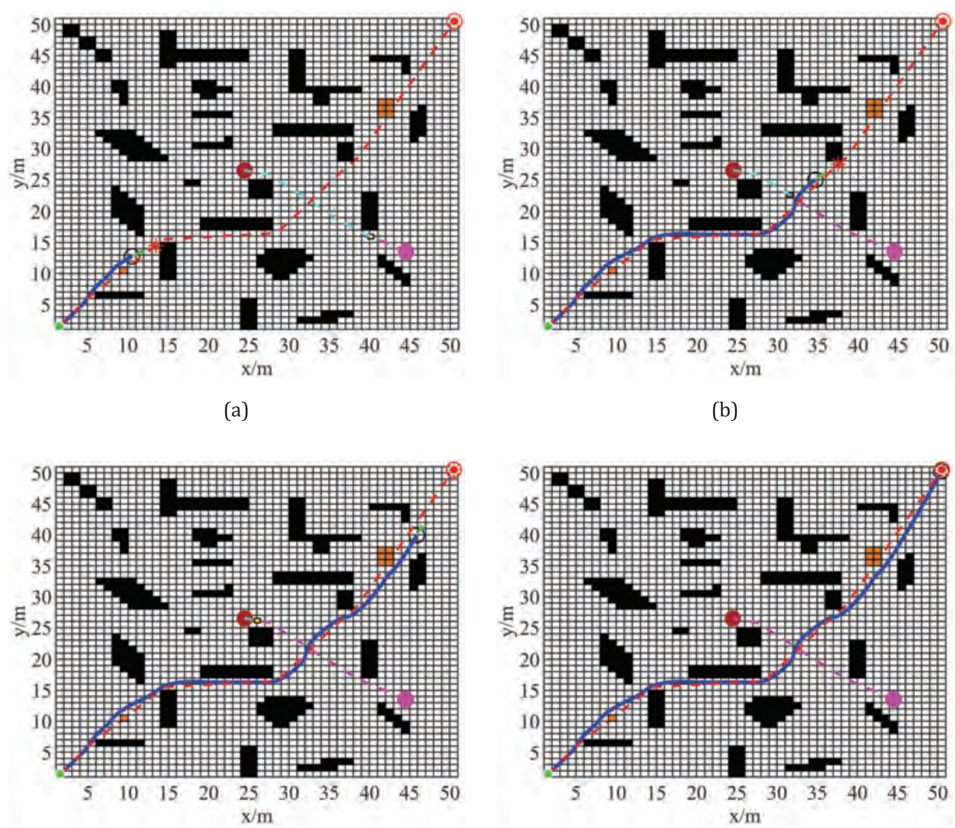
Fig. 16. The path planning trajectories of HHO and DWA in different scenarios: (a) No unknown obstacle scenario, (b) Multiple unknown obstacle scenario.

Table 11. HHO and DWA path planning performance.

Scenario	Algorithm	Path length	Turns	Smoothness	Safety
Figure 14(a)	HHO	75.23	9	No	No
Figure 14(a)	DWA	76.44	8	Yes	Yes
Figure 14(b)	HHO	79.27	10	No	No
Figure 14(b)	DWA	—	—	—	—

were conducted on a 50 × 50 grid map with four different environmental models. A certain number of dynamic obstacles and static unknown obstacles were introduced into the experimental environment. The starting and ending points were represented by green and red dots, respectively. The green fan-shaped region represents the predicted trajectory of AGV. Dynamic obstacles were denoted by yellow small squares, with their starting and ending points marked by purple and brown dots. The planned path was depicted by green dashed lines, while the traversed path was represented by purple dashed lines. Orange rectangles represented unknown obstacles. The trajectory generated by GHHO was shown by red dashed lines, while the trajectory generated by GHHO-DWA fusion algorithm was depicted by blue solid lines.

In Fig. 17, it is evident that AGV progresses along the path planned by GHHO. When encountering unsafe turns, AGV introduces local avoidance to make its path curve safely. When encountering unknown static obstacles, the local avoidance algorithm accurately identifies and avoids them. In the presence of dynamic obstacles, the local avoidance algorithm predicts potential collisions and implements avoidance strategies if necessary. It is clear that AGV swiftly and accurately adopts appropriate avoidance strategies when encountering random unknown obstacles and dynamic obstacles along the global path. Local path planning, through real-time perception and analysis of the surrounding environment, enables AGV to promptly recognize collision risks and react swiftly, selecting smooth and safe paths. This efficient obstacle avoidance capability allows AGV to navigate safely in complex environments, ultimately achieving the intended path planning objectives.



(a) (b) (c) (d)

Fig. 17. Dynamic obstacle avoidance effect of GHHO-DWA algorithm: (a) Avoiding a single position obstacle; (b) Avoiding dynamic obstacles; (c) Avoiding continuous obstacles; (d) Completing dynamic path planning to reach the target point.

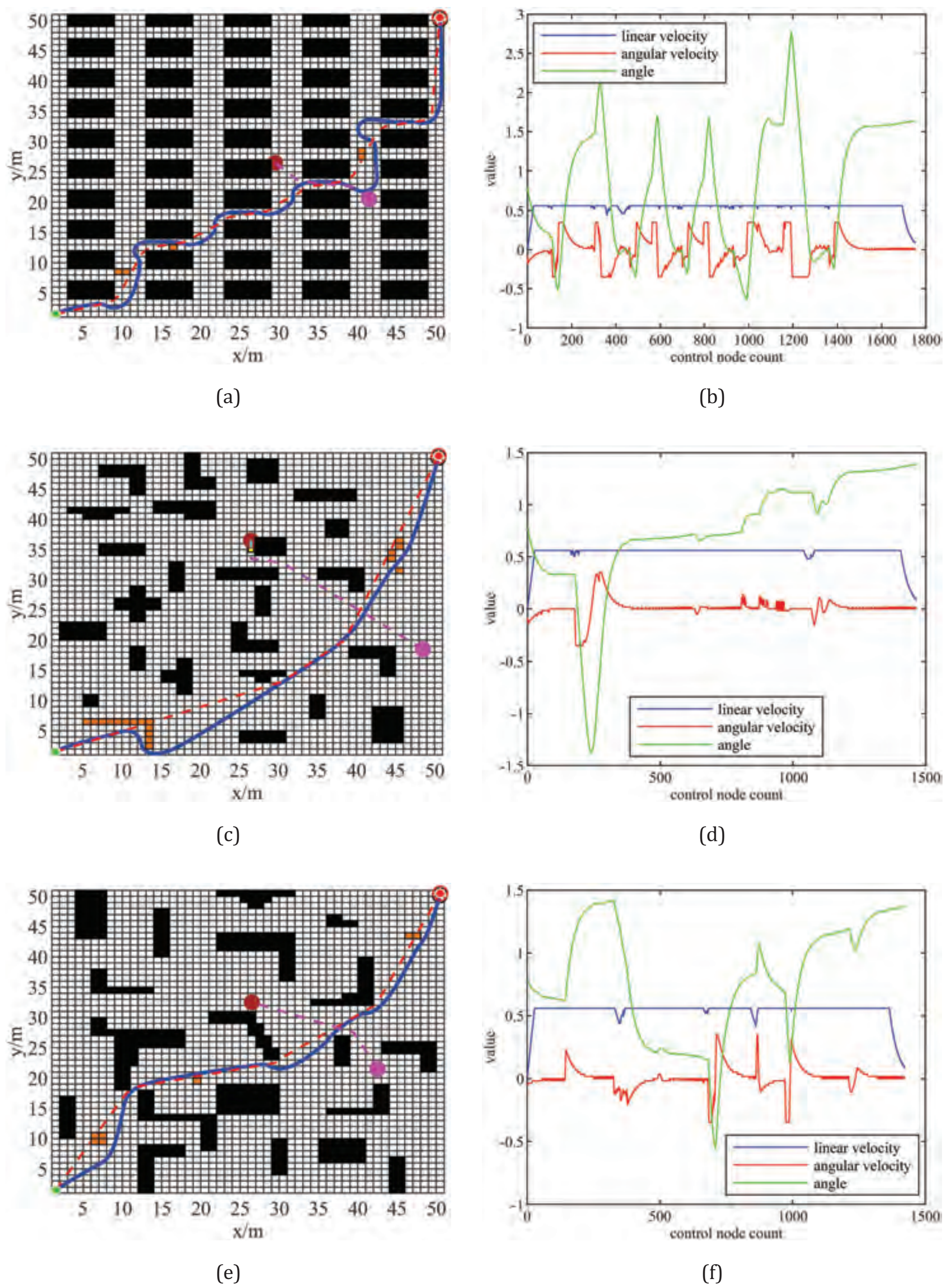
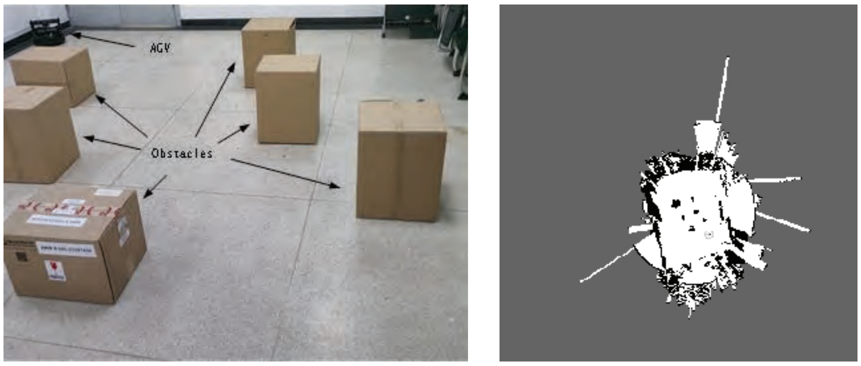


Fig. 18. Fusion algorithm dynamic obstacle avoidance diagram.

To further validate the effectiveness of fusion algorithm, a series of dynamic path planning simulations were conducted in environments of varying complexity levels, comparing the paths generated by fusion algorithm with those determined by GHHO. Through these comparisons, dynamic obstacle avoidance performance of fusion algorithm can be comprehensively evaluated. The comparative results are shown in the Fig. 18.

4.4. *Experimental validation*

To further verify the effectiveness and practicality of algorithm, an indoor AGV path planning experiment was conducted in a real-world environment. This study utilized the QBot 2e, a two-wheeled differential drive mobile robot, equipped with a Raspberry Pi and a 640 × 480 Microsoft Kinect sensor to collect surrounding information. Quanser control model was built using the Simulink functionality in



(a) (b)

Fig. 19. Indoor obstacle environment diagram.

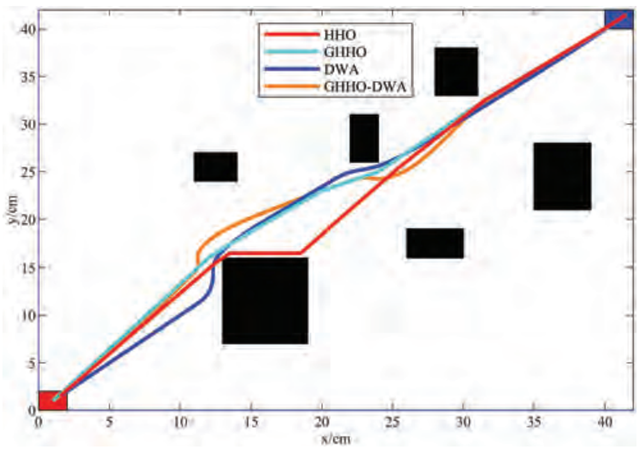


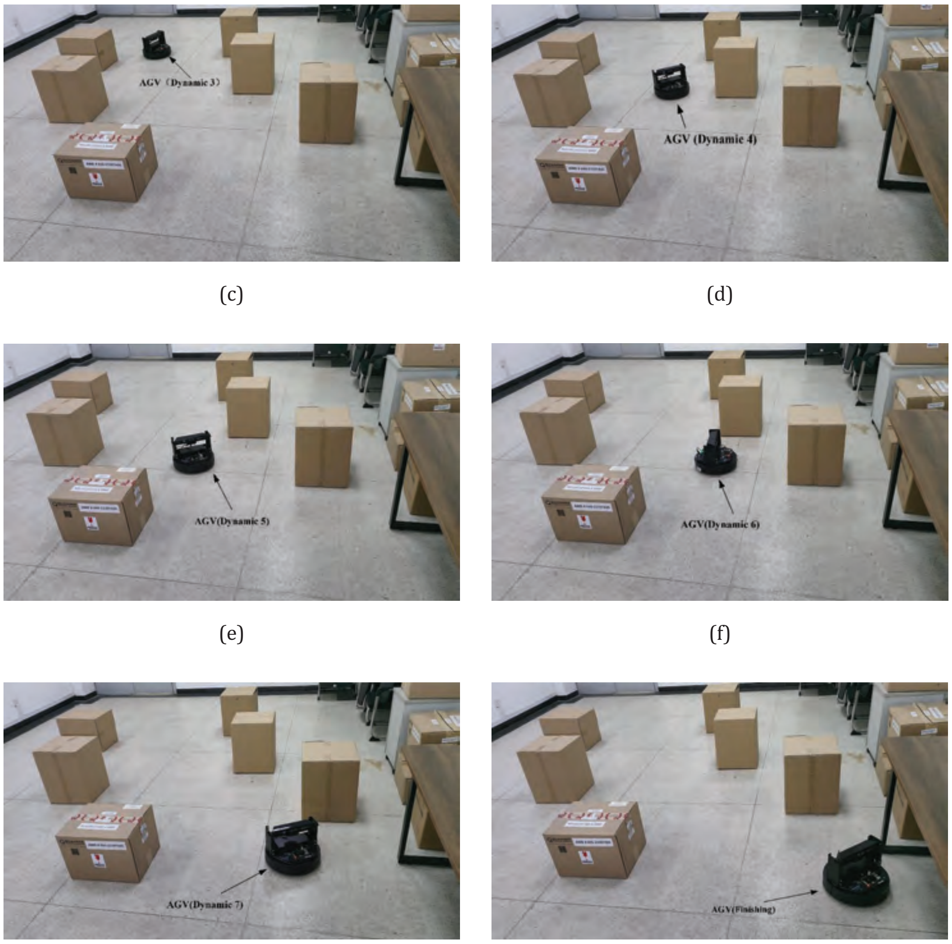
Fig. 20. Validation of path planning algorithm in indoor environment.

the Matlab 2019b environment. The robot was controlled to gather environmental information and scan the generated map, where white represents obstacle-free areas, black represents obstacles, and gray represents unexplored areas. The process algorithm was used to convert the scanned real-world map into a two-dimensional grid map, and the GHHO algorithm, HHO algorithm, DWA algorithm, and GHHO-DWA algorithm were validated on the grid map.

The experimental environment for the indoor AGV path planning experiment in a real-world setting includes six $0.5 \times 0.5 \times 1.0$ cubic meter cardboard boxes. The experimental data parameters are detailed in Sec. 4.3. The necessary code for the experiment is presented in pseudocode form in Secs. 2.2 and 3.2. The dataset used in the experiment consists of the map data generated by the QBot 2e in the aforementioned environment, which is utilized to test the algorithm's performance in complex path planning tasks. The experimental steps are as follows:Establish a connection between the host computer and the QBot 2e robot, and open Quanser control model in Matlab/Simulink environment. Then, control robot to collect data from the surrounding environment and save the collected map to the workspace. Next, process the actual map using the process algorithm, convert it into a two-dimensional grid map format, and define the starting and ending points for the robot. After that, perform path planning in the environment from the previous step to generate a series of continuous points. Finally, provide the path points from the previous step to QBot 2e robot, open the Quanser control model, and conduct algorithm validation.

(a) (b)

Fig. 21. Indoor experimental results display.



(c) (d)

(e) (f)

(g) (h)

Fig. 21. (*Continued*)

In Fig. 19, (a) depicts the distribution of obstacles and the initial position of AGV, while (b) shows the actual map environment scanned by AGV. The process program converts the scanned actual map into a grid map and provides the destination position. The path planning algorithm generates a path, as depicted in Fig. 20. In this figure, the red rectangle represents the starting point of AGV, the blue rectangle represents the destination point of AGV, the black rectangles represent obstacles, and the white areas indicate freely passable regions. The orange solid line represents the path planned by the GHHO-DWA algorithm, the cyan solid line represents the path planned by the GHHO algorithm, the dark blue solid line represents the path planned by the DWA algorithm, and the red solid line represents the path planned by the HHO algorithm. Under the constructed Quanser control model, the generated path points are provided to AGV for path verification. The process of AGV operation is illustrated in Fig. 21.

Table 12. Experimental data of algorithms.

Algorithm	Path length	Turns	Smoothness	Safety
HHO	58.3	3	No	No
DWA	57.41	4	Yes	No
GHHO	57.07	2	No	No
GHHO-DWA	58.13	2	Yes	Yes

Figure 21(a) shows the AGV's starting position and preplanned path; (b) and (c) depict the safe passage through the first obstacle; (d) and (e) illustrate the safe passage through the fourth obstacle; (f) and (g) demonstrate the safe passage through the sixth obstacle; and (h) shows the AGV reaching its destination.

From Table 12 and Fig. 20, it is clear that although the GHHO and DWA algorithms generate shorter paths, they often result in unreachable paths in practical applications, leading to collisions with obstacles. In contrast, while the GHHO-DWA algorithm produces slightly longer paths, these paths are reachable and smooth. Experimental validation shows that, considering the AGV's radius, GHHO-DWA offers feasible and safe paths with the shortest safe distance. In summary, GHHO-DWA integrates smoothness, reachability, and safety in path planning, demonstrating significant overall advantages and significantly reducing collision risks for the QBot2e during movement. The experimental results confirm the effectiveness and practicality of the GHHO-DWA algorithm, and its validity has been verified using the QBot2e platform.

5. Conclusions and Future Works

Based on the improved HHO and DWA algorithms, this paper proposes a new optimal path planning algorithm for indoor AGVs called GHHO-DWA. First, the improved HHO algorithm is used to generate the global optimal path, with its key nodes serving as local target points for the DWA algorithm. Then, the improved DWA algorithm is employed for local path planning, ensuring that the AGV reaches the target point safely and smoothly. The effectiveness of the GHHO-DWA algorithm is validated through targeted simulations, and its practicality is further confirmed through experiments on the QBot2e platform.

The experimental results show that the GHHO-DWA algorithm can effectively generate reachable and smooth paths in various obstacle environments, a key advantage that highlights its significant potential for application in complex indoor environments. However, due to limitations in the experimental equipment, the algorithm's performance in environments with random and dynamic obstacles has not yet been fully tested, which may affect its adaptability and path replanning capabilities in dynamic settings. Additionally, the algorithm's computational complexity could become a bottleneck in larger-scale or more complex environments, potentially limiting its real-time performance and efficiency.

Despite these challenges, the GHHO-DWA algorithm continues to demonstrate outstanding adaptability and practical value in handling complex path planning tasks. Future research should focus on the following areas: first, further testing the algorithm's performance in random and dynamic obstacle environments to verify its robustness and adaptability in real-world applications; second, validating the algorithm's generalizability across a wider range of AGV platforms to ensure its effectiveness in various practical scenarios; and finally, optimizing the algorithm's computational complexity to enhance its real-time performance and collaborative capabilities in complex environments, thereby better meeting the demands of practical applications. These efforts will further enhance the competitive advantage of the GHHO-DWA algorithm in dynamic environments and expand its potential for industrial application.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China under Grant (62103127, 62373132), Youth Natural Science Foundation of Hebei Province under Grant F2020201048, Funded by Science and Technology Project of Hebei Education Department under Grant BJK2023057, Advanced Talents Incubation Program of the Hebei University under Grant 521000981366, The Innovation Capability Improvement of Hebei Province 22567629H and The Excellent Youth Research Innovation Team of Hebei University under Grant QNTD202411.

References

[1] S. Aggarwal and N. Kumar, Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, *Computer Commun.* **149** (2020) 270–299.

[2] Y. Hou *et al.*, Multi-agent path finding for non-conflicting paths: An infinite speed conflict-based search approach, in *Proc. 42nd Chinese Control Conf.* (IEEE, 2023), pp. 5500–5505.

[3] Z. Zang *et al.*, A coordinated behavior planning and trajectory planning framework for multi-UGVs in unstructured narrow interaction scenarios, *IEEE Trans. Intell. Veh.* **1** (2024) 1–14.

[4] T. Ong *et al.*, Multi-drone optimal mission assignment and 3D path planning for disaster rescue, *Drones* **7**(6) (2023) 394.

[5] Y. Zhang *et al.*, Multi-AGV path planning for indoor factory by using prioritized planning and improved ant algorithm, *J. Eng. Technol. Sci.* **50**(4) (2018) 534–547.

[6] Z. Zhang *et al.*, Energy-efficient path planning for a single-load automated guided vehicle in a manufacturing workshop, *Comput. Ind. Eng.* **158** (2021) 107397.

[7] L. Xu, W. Ning and L. Xufeng, Study on conflict-free AGVs path planning strategy for workshop material distribution systems, *Procedia CIRP* **104** (2021) 1071–1076.

[8] J. A. Douthwaite, S. Zhao and L. S. Mihaylova, Velocity obstacle approaches for multi-agent collision avoidance, *Unmanned Syst.* **7**(1) (2019) 55–64.

[9] Z. Zang *et al.*, A unified framework integrating trajectory planning and motion optimization based on spatio-temporal safety corridor for multiple AGVs, *IEEE Trans. Intell. Veh.* **9**(1) (2023) 1217–1228.

[10] K. Karur *et al.*, A survey of path planning algorithms for mobile robots, *Vehicles* **3**(3) (2021) 448–468.

[11] Y. K. Ever, Using simplified swarm optimization on path planning for intelligent mobile robot, *Procedia Comput. Sci.* **120** (2017) 83–90.

[12] C. Sahu, P. B. Kumar and D. R. Parhi, An intelligent path planning approach for humanoid robots using adaptive particle swarm optimization, *Int. J. Artif. Intell. Tools* **27**(5) (2018) 1850015.

[13] Y. He and M. Wang, An improved chaos sparrow search algorithm for UAV path planning, *Sci. Rep.* **14**(1) (2024) 366.

[14] C. Cai *et al.*, A path planning method using modified Harris Hawks optimization algorithm for mobile robots, *PeerJ Comput. Sci.* **9** (2023) e1473.
[15] G. Li *et al.*, Robot path planning based on improved Harris Hawks optimization algorithm, in *Proc. 5th Int. Conf. Intell. Auton. Syst.* (IEEE, 2022), pp. 209–214.
[16] J. Gregoriȅ, M. Seder and I. Petroviȅ, Autonomous hierarchy creation for computationally feasible near-optimal path planning in large environments, *Robot. Auton. Syst.* **172** (2024) 104584.
[17] F. Rodríguez *et al.*, Collision-free path planning for multiple robots using efficient turn-angle assignment, *Robot. Auton. Syst.* **104** (2024) 104698.
[18] A. A. Dehkordi *et al.*, Nonlinear-based chaotic Harris Hawks Optimizer: Algorithm and internet of vehicles application, *Appl. Soft Comput.* **109** (2021) 107574.
[19] A. Mohammadzadeh and B. Firouzi, A new path following scheme: Safe distance from obstacles, smooth path, multi-robots, *J. Ambient Intell. Human Comput.* **14**(4) (2023) 4621–4633.
[20] H. Wu *et al.*, A novel fuzzy control path planning algorithm for intelligent ship based on scale factors, *J. Supercomput.* **80**(1) (2024) 202–225.
[21] Z. Lin *et al.*, Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles, *Trans. Inst. Meas. Control* **44**(1) (2022) 121–132.
[22] Z. Zhou *et al.*, A hybrid algorithm for dynamic path planning of mobile robot using improved A* and DWA, in *Proc. Int. Conf. Mech. Des.* (Springer, Singapore, 2023), pp. 2483–2503.
[23] M. Ding *et al.*, Collision-free path planning for cable-driven continuum robot based on improved artificial potential field, *Robotica* **42**(5) (2024) 1350–1367.
[24] P. Marin-Plaza *et al.*, Global and local path planning study in a ROS-based research platform for autonomous vehicles, *J. Adv. Transp.* **2018**(1) (2018) 6392697.
[25] H. Zhang, W. Lin and A. Chen, Path planning for the mobile robot: A review, *Symmetry* **10**(10) (2018) 450.
[26] N. I. Bouaziz *et al.*, Trajectory planning for autonomous formation of wheeled mobile robots via modified artificial potential field and improved PSO algorithm, *Unmanned Syst.* **12**(6) (2024) 1–20.
[27] A. A. Heidari *et al.*, Harris Hawks Optimization: Algorithm and applications, *Future Gener. Comput. Syst.* **97** (2019) 849–872.
[28] M. Radmanesh *et al.*, Overview of path-planning and obstacle avoidance algorithms for UAVs: A comparative study, *Unmanned Syst.* **6**(2) (2018) 95–118.
[29] L. Wu *et al.*, Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot, *Expert Syst. Appl.* **215** (2023) 119410.
[30] K. Dong *et al.*, Dynamic planning method of evacuation route in dam-break flood scenario based on the ACO-GA hybrid algorithm, *Int. J. Disaster Risk Reduct.* **100** (2024) 104219.
[31] L. Liu *et al.*, Dynamic path planning of mobile robot based on improved sparrow search algorithm, *Biomimetics* **8**(2) (2023) 182.

Zhen Zhou received his Ph.D. degree from Automation Department of Yanshan University in 2018. He is currently Professor at the College of Electronic and Information Engineering, the Hebei University, Baoding, China. His research interests include multiple robotic systems, distributed optimization theory and machine vision.

Yujie Song received his Bachelor's degree in automation from the North China University of Science and Technology in 2023. He is currently Master's student in the Department of Control Engineering, the Hebei University, Baoding, China. The main research direction is the path planning of indoor AGV.

Shuo Li received his Bachelor's degree in Robotics Engineering from Luoyang Institute of Science and Technology,China. He is currently a master's student in the Department of control engineering, Hebei University, China. His research interests include path planning for dual-arm robots.