ISSN: 2688-8645

An Integrated Global-Local Path Planning Framework for Indoor Mobile Robots in Structured Environments

Jiangchao Zhang, Lei He *, Zhenfeng Lin, Yile Sang and Rongrong Ma

School of Electrical Engineering, Southwest Minzu University, Chengdu, Sichuan, 610041, China

* Corresponding author: Lei He

Abstract

To address the challenges of balancing path quality, search efficiency, and dynamic obstacle avoidance for mobile robots in structured library environments, a global–local collaborative path planning method is proposed. First, an improved A* algorithm is employed to generate a global path. Subsequently, an improved Hippopotamus Optimization (HO) algorithm is introduced for path optimization, and local planning is integrated to achieve dynamic obstacle avoidance. Simulation results demonstrate that, compared with the conventional A* algorithm, the improved A* algorithm reduces the number of searched nodes by 20.00% and computation time by 12.50%. Compared with the original HO algorithm, the improved HO algorithm reduces path length by 15.62% and 17.93% in the bookshelf and lounge areas, respectively, while decreasing execution time by 31.35% and 28.96%, respectively, with a 25% reduction in turning points in both areas. Physical robot experiments further validate the navigation capabilities of the proposed method in scenarios involving static obstacles, dynamic pedestrians, and U-shaped corridors. The proposed method effectively integrates global path optimization with local dynamic obstacle avoidance, thereby improving the overall performance of mobile robot path planning in structured indoor environments.

Keywords

Indoor Mobile Robots; Global-Local Path Planning; Hybrid Optimization; Obstacle Avoidance; Library Environments.

1. Introduction

Path planning in structured indoor environments remains challenging due to constrained passages, dense obstacles, and static-dynamic obstacles, requiring balance between path optimality, efficiency, adaptability, and global-local coordination.

Library environments represent typical structured indoor scenarios with dense obstacles, narrow passages, and human-induced disturbances **(Fig. 1)**.



(a) U-shaped passage formed by bookshelves (b) Mixed static-dynamic obstacles

Fig 1. Schematic illustration of a representative structured indoor environment

ISSN: 2688-8645

For global path planning in structured indoor environments, classical graph-based methods, such as Dijkstra and A*, are widely adopted. Dijkstra guarantees shortest paths but may be inefficient with redundant turns [1-2] (Zhu et al., 2021; Zhang et al., 2020), whereas A* improves search efficiency through heuristic guidance but may still generate excessive inflection points in constrained layouts (Zhang et al., 2024; Tang et al., 2021) [3-4]. Sampling-based methods, such as Rapidly-exploring Random Trees (RRT), obtain feasible solutions in high-dimensional obstacle-dense spaces; however, their trajectories are often irregular and computationally expensive, limiting suitability for smooth navigation in structured environments (Li et al., 2023; Zhang et al., 2023; Chen et al., 2021) [5-7]. Therefore, swarm-based metaheuristic algorithms have been increasingly explored to enhance global optimization and path refinement, including HO (Amiri et al., 2024) [8], GWO (Mirjalili et al., 2014) [9], SSA (Mirjalili et al., 2017) [10], and PSO (Kennedy and Eberhart, 1995) [11]. Amiri et al. (2024) [8] compared HO with GWO, SSA, and PSO on CEC2005 benchmark functions, where HO achieved the best performance with the lowest Friedman mean rank. Representative results on F1, F9, and F22 are summarized in **Table 1**.

Table 1 Performance summary of several typical intelligent optimization algorithms

Algorithm	Representative results (Mean (Std))	Friedman mean rank on F1–F23	Computational complexity
Hippopotamus Optimization	F1: 0 (0); F9: 0 (0); F22: -10.403 (6.16E-05)	2.2468	O(Nm(1 + 5T/2))
Gray wolf optimization	F1: 5.73E-88 (1.78E-87); F9: 4.9095 (8.2235); F22: -9.5722 (2.1838)	4.5300	O(Nm(1 + T))
Salp Swarm Algorithm	F1: 2.71E-07 (4.40E-07); F9: 52.899 (17.219); F22: -8.8874 (2.8452)	4.8758	O(Nm(1 + T))
Particle Swarm Optimization	F1: 3.69E-06 (1.19E-05); F9: 45.735 (14.675); F22: -6.7572 (3.7466)	6.3291	O(Nm(1 + T))

Note: F1, F9, and F22 represent unimodal, high-dimensional multimodal, and fixed-dimensional multimodal functions, respectively. Results are reported as Mean (Std), and lower Friedman ranks indicate better performance. N, m, and T denote the population size, problem dimensionality, and maximum number of iterations, respectively.

Local planners provide real-time motion adjustment around dynamic obstacles. The artificial potential field (APF) method guides robot motion through attractive forces from the target and repulsive forces from obstacles (Song et al., 2020; Liu et al., 2022) [12, 13]. Although simple and responsive, APF is prone to local minima in U-shaped obstacles or narrow passages when forces counteract. The dynamic window approach (DWA) evaluates feasible velocity commands through short-horizon simulation (Xu et al., 2023; Lai et al., 2023) [14, 15], but limited global awareness may cause local optima and reduced trajectory smoothness in complex indoor environments. Therefore, effective global-local coordination is essential for reliable navigation

ISSN: 2688-8645

under mixed static and dynamic constraints. To address this challenge, this paper proposes an integrated global-local path planning framework.

The remainder of this paper is organized as follows: Section 2 presents the overall framework of the proposed hybrid path planning method. Section 3 describes the improved global planning strategy based on A* and Hippopotamus Optimization and evaluates its performance through simulations. Section 4 introduces the local path planning strategy for navigation in confined passages. Section 5 reports experimental results obtained in structured indoor environments with static and dynamic obstacles. Finally, Section 6 gives conclusions.

2. Integrated Global-Local Path Planning Framework

Structured indoor environments, such as library book rooms, feature dense static structures and dynamic obstacles caused by human activities, imposing simultaneous requirements on global path feasibility and local real-time responsiveness. To address these challenges, an integrated global-local path planning framework is developed that coordinates global planning and local reactive strategies within a unified architecture. The overall framework is illustrated in **Fig. 2**.

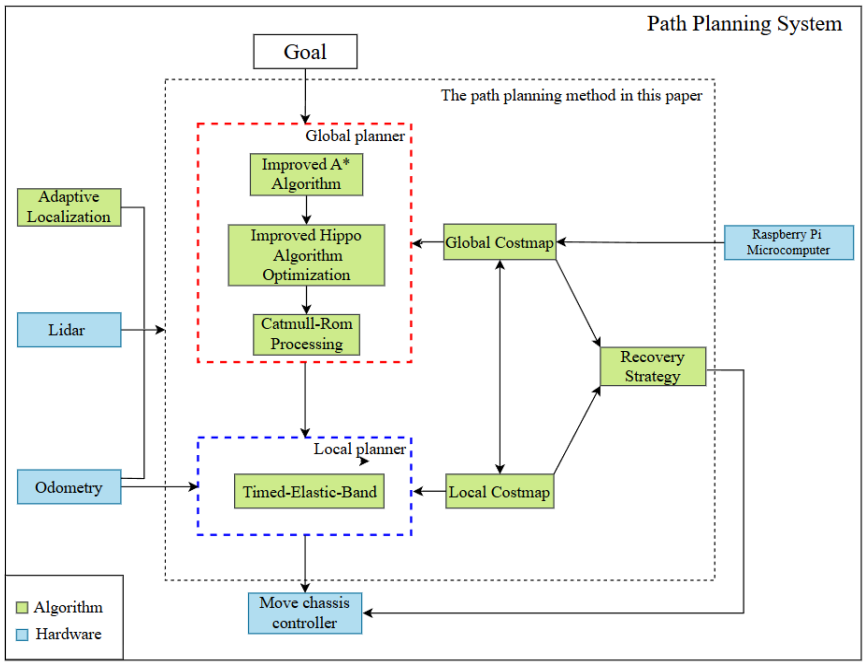


Fig 2. Overall structure of the integrated global-local path planning framework

The proposed framework consists of two tightly coupled components: global path planning and local path planning.

For global planning, an improved A* algorithm is first provides deterministic guidance and generate an initial feasible path. To further improve path quality while preserving global feasibility, an enhanced Hippopotamus Optimization (HO) algorithm is introduced to refine the deterministic path, leveraging its balanced global exploration and local exploitation capability. The resulting path nodes are then used as structured prior information for refinement, improving path smoothness and overall quality under complex layout constraints. Catmull-Rom spline smoothing is applied to ensure geometric feasibility of the refined global path.

For local planning, a Timed Elastic Band (TEB)-based method is adopted to handle dynamic obstacle avoidance and short-term motion feasibility. By incorporating temporal constraints and trajectory deformation, the local planner adjusts the navigation trajectory in real time while

ISSN: 2688-8645

maintaining consistency with the global reference path, enabling effective navigation in narrow passages and dynamically changing indoor environments.

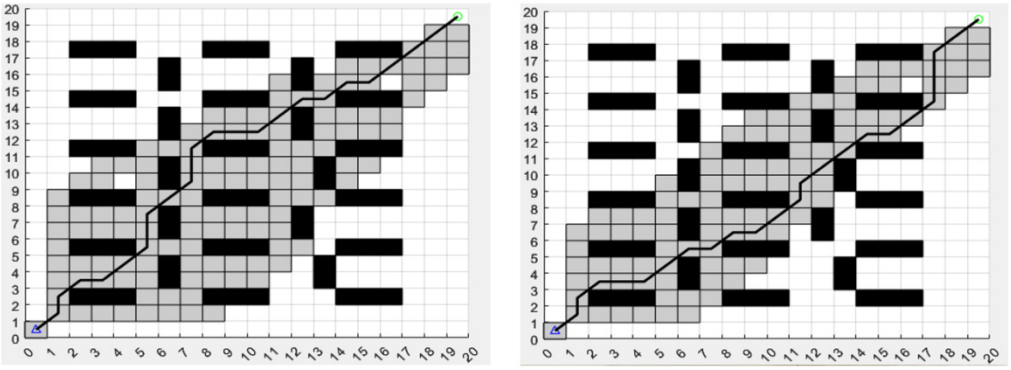
3. Global Planning via Deterministic Search and Heuristic Optimization

3.1. Deterministic Global Decision-Making (A*)

At the global level, an improved A* algorithm is adopted as the deterministic planner to generate a feasible path considering obstacle distribution and passage structures. Its heuristic function $h(n)$ estimates the cost from the current node to the goal, and its design directly influences search efficiency and path quality. However, in structured indoor environments with narrow passages and dense obstacles, conventional A* often produces paths with excessive inflection points and limited smoothness. To address this issue, an adaptive dynamic weighting method is introduced, as shown in **Eq. (1).**

$$\begin{cases} h(n) = \omega(n) \cdot \left(\alpha d_E + \beta d_M + \gamma d_D\right) \\ \omega(n) = \exp\left(\frac{d_E}{D_E}\right) - \frac{d_E}{D_E} \\ D_E = \sqrt{\left(x_{origin} - x_{goal}\right)^2 + \left(y_{origin} - y_{goal}\right)^2} \end{cases} \tag{1}$$

where n denotes the current node, $\omega(n)$ is the dynamic heuristic factor, and d_E , d_M , and d_D represent Euclidean, Manhattan, and diagonal distances from the current node to the goal node, respectively. D_E denotes the Euclidean distance from the start node to the goal node. The parameters α , β , and γ represent the weighting coefficients of the three heuristic distance components, respectively. These coefficients satisfy the normalization constraint $\alpha+\beta+\gamma=1$.



(a) Traditional A* algorithm (b) The Improved A* algorithm

Fig 3. Simulation results of two algorithms

The dynamic factor $\omega(n)$ is determined by the normalized distance ratio $\frac{d_E}{D_E}$. At the beginning of the search process, the current node coincides with the starting node, resulting in $d_E = D_E$. Therefore, the initial value of the dynamic factor during the search process is $\omega(n) = e - 1 \approx 1.718$. As the search progresses, d_E gradually decreases as the current node approaches the goal node. Since $0 \le \frac{d_E}{D_E} \le 1$, the dynamic factor satisfies $1 \le \omega(n) \le 1.718$. A larger $\omega(n)$ when the node is farther from the goal strengthens heuristic guidance and improves search efficiency, whereas a smaller $\omega(n)$ near the goal reduces the search step size and facilitates convergence.

ISSN: 2688-8645

The performance of the improved A* algorithm with adaptive dynamic weighting was evaluated against the conventional A* algorithm in a simulated library environment implemented in MATLAB, as shown in **Fig. 3**.

The traditional A* algorithm produced a path length of 29.799 m with a calculation time of 0.136 s and 130 traversed nodes, whereas the improved A* reduced these values to 29.376 m, 0.119 s, and 104 nodes, respectively. These results demonstrate that the adaptive dynamic weighting strategy improves path quality and search efficiency.

3.2. Heuristic Global Path Refinement Using Improved Hippopotamus Optimization

Building upon the deterministic global path obtained in Section 3.1, a heuristic optimization layer is introduced to improve path smoothness and node distribution, thereby enhancing global path quality while preserving feasibility. HO is adopted because its explicit exploitation phase enables local refinement around promising solutions, making it suitable for refining control points along the deterministic path and linking global planning with subsequent local planning. To further enhance optimization robustness, chaotic mapping and Lévy flight strategies are integrated to improve population diversity, strengthen global search capability, and mitigate premature convergence. The detailed optimization procedure of the improved HO algorithm is described below.

3.2.1. Population Initialization

Randomly generate an initial hippo population (a set of candidate solutions) within the problem solution space, using the **Eq. (2).**

$$X_{ij} = lb_j + r(N) \cdot (ub_j - lb_j) \tag{2}$$

Where X_{ij} represents the position of the i-th hippo (the i-th candidate solution) in the j-th dimension. This study focuses on the path position coordinates of a robot in a plane, thus the problem-solving dimension j=2 (including x-coordinate and y-coordinate). lb_j and ub_j represent the lower and upper bound of the decision variable in the j-th dimension, thereby defining the range of the search space. For example, in the simulation map of this paper, $lb_x = 0$, $ub_x = 20$, $lb_y = 0$, $ub_y = 20$. $r(N) = rand(N,1)$, it represents an N-by-1 column vector, whose elements are uniformly distributed random numbers in the range [0, 1], N denotes the population size. This ensures the initial population is uniformly distributed across the two-dimensional search plane.

The original hippo algorithm uses a uniform distribution to generate the initial population, which may lead to slow convergence speed and low search efficiency. To address this issue, this paper introduces a Logistic chaotic map for improvement(Alawida et al.,2024) [16]. During initialization, it can generate highly irregular and well-distributed numerical sequences, thereby better simulating biological populations with high diversity, wide dispersion, and complex dynamic characteristics. The mathematical model of the Logistic chaotic map is given by **Eq.(3).**

$$z_{i+1}^N = \mu z_i^N (1 - z_i^N), \ \mu \in [0,4], \ z_i^N \in (0,1) \tag{3}$$

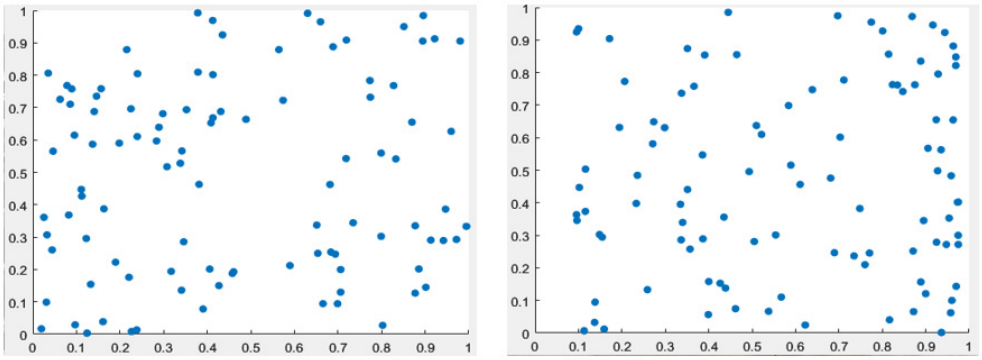
Where K denotes the population size, i is the iteration count, and μ is the Logistic parameter with ranges from 0 to 4.

The initialization process is modified by substituting a Logistic chaotic map for the random uniform distribution. The improved equation is given by:

$$X_{ij} = lb_j + z_{i+1}^N \cdot (ub_j - lb_j) \tag{4}$$

ISSN: 2688-8645

Random uniform distribution initialization and Logistic chaotic map initialization were compared in MATLAB simulations, the results shown in **Fig. 4** (population size $N = 100$, Logistic parameter $\mu = 3.9$, $z_1^{100} = 0.1$, $lb_x = 0$, $ub_x = 1$, $lb_y = 0$, $ub_y = 1$).



(a) uniform distribution initialization (b) Logistic chaotic map initialization

Fig 4. Comparison of different initialization methods

As shown in **Fig. 4(b)**, Logistic chaotic initialization generates a more uniform population distribution, thereby improving global exploration capability and reducing the risk of convergence to local optima.

3.2.2. Exploration Phase (Global Search)

This stage simulates the social interactions within the hippo group and the curiosity of young hippos, aiming to conduct global exploration. The exploration phase is further divided into two sub-parts: hippo group position update (seeking advantageous positions, i.e., optimal solutions) and defense against predators (enhancing the ability to escape local optima).

(i) Hippo group position update

(a) Male hippo position update: Male individuals move toward the hippo leader (the current optimal solution), compete for dominant positions, and attempt to replace it. This process can be expressed as follows:

$$X_{male}^{i+1} = X_{male}^{i} + r_{male} \cdot (X_{best}^{i} - I_1 \cdot X_{male}^{i}) \tag{5}$$

In this study, the iteration index i starts from 1 and increases sequentially during the optimization process, while T denotes the maximum number of iterations.

Where X_{male}^{i+1} represents the updated position of the male hippo, and X_{male}^{i} represents its position at the i-th iteration. X_{best}^{i} represents the position of the hippo with the best fitness in the population at the i-th iteration, which is determined by finding the minimum value of the total path cost (the objective function). r_{male} represents a random scaling factor for the search step size, which denotes a uniformly distributed random number in the interval [0,1] . It ensures that even for movements in the same direction, the step length is not identical each time, thereby increasing search diversity and avoiding patterned movement. I_1 represents a search direction control factor, which governs the degree of convergence toward the optimal individual, taking a random integer value of either 1 or 2. When the value is 1, it indicates a stronger tendency to approach the dominant individual (the current optimal solution), demonstrating a preference to explore known high-quality regions. If the value is 2, it indicates the individual is driven away by the hippo leader, showing a greater tendency to explore unknown regions in search of potentially better solutions and to avoid premature convergence to local optima.

To enhance the convergence speed of the aforementioned algorithm, this paper introduces an improvement by setting an adaptive step size coefficient $\omega(i)$. The improved algorithm is as follows:

ISSN: 2688-8645

$$\begin{cases} X_{male}^{i+1} = X_{male}^{i} + \omega(i) \cdot r_{male} \cdot (X_{best}^{i} - I_1 \cdot X_{male}^{i}) \\ \omega(i) = \sqrt{\exp(1 - \frac{i}{n})} \end{cases} \tag{6}$$

In the formula, $\omega(i)$ is the weight coefficient, i is the current iteration count, and n is the set maximum iteration count.

In the early stages of iteration, the step size coefficient is relatively large, enabling the algorithm to rapidly search near the relatively optimal solution. As the number of iterations increases, the step size coefficient gradually decreases, allowing for more refined local search around these near-optimal solutions.

(b) Female/juvenile hippo position update: Juvenile hippos exhibit strong curiosity and active exploration in their behavioral patterns. Their actions can be categorized into two scenarios: exploring away from the group and following the group for safety. Female hippos primarily follow behavioral patterns centered around protecting their young and staying with the herd. Their positions both converge toward the optimal solution and maintain relatively stable locations (suitable for local exploration).

The equations for updating the position of female hippos are defined as:

$$X_{female}^{i+1} = \begin{cases} X_{female}^{i} + r_{female}(X_{best}^{i} - I_2 X_{female}^{i}) \quad if \quad r_{female} > 0.5 \quad ① \\ X_{female}^{i} + r_{female}(X_{best}^{i} - I_2 \cdot MeanGroup) \quad otherwise \quad ② \end{cases} \tag{7}$$

Among them, 1) denotes the following mode, and 2) denotes the group influence mode. X_{female}^{i+1} represents the updated position of the female hippo, and X_{female}^{i} represents its position at the i-th iteration. The learning factor r_{female} takes its value as a uniformly distributed random number in [0, 1]. I_2 as an inertia factor, randomly takes the value of 1 or 2, just like I_1. MeanGroup represents the average position of a random subgroup, as follows:

$$MeanGroup = \frac{1}{K} \sum_{k=1}^{K} X_{subk} \tag{8}$$

Where K is the size of the random subgroup (typically 2-5 individuals), and X_{subk} represents the positions of the individuals randomly selected for the subgroup (excluding the individual itself and the globally optimal position).

The equations for updating the position of juvenile hippos are defined as:

$$X_{\text{young}}^{i+1} = \begin{cases} X_{young}^{i} + A \cdot (X_{best}^{i} - I_3 \cdot MeanGroup) \ \ if \ T > 0.6 \quad ① \\ X_{young}^{i} + B \cdot (X_{best}^{i} - I_4 \cdot X_{young}^{i}) \ \ if \ T \le 0.6 \quad ② \end{cases} \tag{9}$$

The first update mode promotes global exploration during the early iterations, whereas the second mode emphasizes local exploitation during the later iterations.

Where X_{young}^{i+1} represents the updated position of the young hippo, and X_{young}^{i} represents its position at the i-th iteration. I_3 and I_4 similarly represent random integers, each taking the value of 1 or 2. A represents the exploration intensity coefficient, and B denotes the exploitation intensity coefficient, which are expressed as:

$$\begin{cases} A = 2 \cdot rand \cdot (1 - \frac{i}{n}) \\ B = rand \cdot (\frac{i}{n}) \end{cases} \tag{10}$$

ISSN: 2688-8645

Where i represents the current iteration number, n represents the maximum number of iterations.

The factor T decreases with increasing iteration count i and is used to balance exploration and exploitation, as expressed in **Eq. (11).**

$$T = e^{(-\frac{i}{n})} \tag{11}$$

As the iteration proceeds, the adaptive factor T gradually shifts the search from global exploration toward local exploitation.

(ii) Defense against predators (enhancing the ability to escape local optima)

The predator's position is defined by the following expression:

$$P_j = lb + z_p^M \times (ub - lb) \tag{12}$$

Where j represents the j-th predator, and P_j denotes the position of the j-th predator generated. z_p^M represents an M-by-1 column vector, whose elements are random numbers generated using Logistic chaotic mapping, expressed in **Eq. (13)**. M denotes the population size.

$$z_p^M = \mu z_p^M (1 - z_p^M),\ \ \mu \in [0,4],\ \ z_p^M \in (0,1) \tag{13}$$

Then, the distance from the j-th predator to the q-th hippo can be expressed as:

$$d_{jq} = \left| P_j - X_q \right| \tag{14}$$

Where X_q represents the position of the q-th hippo.

Hippo defensive behavior equation:

$$X_{q,hippo} = \begin{cases} RL \cdot P_j + \frac{b}{c - d \cdot \cos(g)} \cdot \frac{1}{d_{jq}} & (when\ f_{predator,j} < f_q) \\ RL \cdot p_j + \frac{b}{c - d \cdot \cos(g)} \cdot \frac{1}{2 d_{jq} + r_{defense}} & (when\ f_{predator,j} \geq f_q) \end{cases} \tag{15}$$

RL is a random vector following a Lévy distribution, which is characterized by occasional long jumps. This property models the sudden position shifts of a predator attacking a hippopotamus, thus effectively helping the algorithm escape local optima. b, c, and d are uniformly distributed random numbers ($b \in [2,4]$, $c \in [1,1.5]$, $d \in [2,3]$). g is a random number between - 2π and 2π . $r_{defense}$ is a uniformly distributed random number within the closed interval [-1,1]. $f_{predator,j}$ is the fitness value of the predator, f_q is the fitness value of the q-th hippopotamus. If $f_{predator,j} < f_q$, the predator is considered closer; otherwise ($f_{predator,j} \geq f_q$), it is considered farther away.

3.2.3. Exploitation Phase (Local Search)

When a hippopotamus fails to repel predators, it moves toward the nearest safe region, which corresponds to a localized search process. This mechanism endows the HO algorithm with effective local exploitation capability, enabling fine-grained refinement of solutions and facilitating smooth transition to subsequent local planning.

(i) Local search range update

Based on the current iteration number i and the maximum iteration number n, the lower and upper bounds of the local search are dynamically adjusted.

$$\begin{cases} lb_{local} = \dfrac{lb}{i} \\ ub_{local} = \dfrac{ub}{i} \end{cases} \tag{16}$$

Where lb and ub are the lower and upper bounds of the global search space, lb_{local} and ub_{local} define the lower and upper bounds of the local search space during the current iteration.

ISSN: 2688-8645

(ii) Position update of a hippopotamus

$$X_{safety,i} = X_i + r_{local}\left[lb_{local} + d \cdot (ub_{local} - lb_{local})\right]$$
$$d = \begin{cases} 2 \times \vec{r}_1 \\ r_2 \\ r_3 \end{cases} \quad (17)$$

Where $X_{safety,i}$ represents the updated position generated by the hippopotamus in order to locate a safer area. r_{local} is a random number uniformly distributed in [0,1], used to control the search step size. The parameter d functions as a mechanism to diversify the local search, implemented by randomly selecting a value or vector from three predefined scenarios. $\vec{r}_1$ represents a random vector in [0,1], which simulates the random observation and imitation behavior among individuals. It disrupts the absolute reliance on the single information source (X_{best}), redirecting the search toward other potential regions of the solution space. r_2 is a uniformly distributed random number in [0,1], which simulates a fine-grained search in the neighborhood of the optimal solution to locate potentially better points. r_3 is a random number drawn from the standard normal distribution, i.e., $r_3 \sim N(0,1)$ and is used to introduce Gaussian perturbations during the local search.

3.2.4. Performance Evaluation and Comparison

(1) Benchmark-based comparison on CEC2005 (SSA/HO / Improved HO)

To assess the optimization robustness of the improved Hippopotamus Optimization (HO) algorithm, benchmark experiments were conducted using representative test functions from the CEC2005 suite. Both unimodal and multimodal functions were selected to evaluate local exploitation capability and global exploration behavior. All experiments were conducted in a MATLAB-based simulation environment. The mean and standard deviation (Mean ± Std) of the final fitness values obtained by SSA, the original HO, and the improved HO over 50 independent runs are reported in **Table 2(a)**, where the improved HO achieved the best or tied-best mean performance on all six benchmark functions.

The non-parametric test results are summarized in **Table 2(b)**. For each benchmark function, the Kruskal–Wallis test was used to examine overall differences among SSA, the original HO, and the improved HO. Pairwise comparisons between the improved HO and the competing algorithms were then conducted using the Wilcoxon rank-sum test, with Holm correction applied to the 12 comparisons.

The Kruskal-Wallis test showed significant differences among the three algorithms on all six functions. After Holm correction, improved HO significantly outperformed SSA on all functions and HO on F2, F4, F8, and F15, confirming the effectiveness of the proposed improvements. Therefore, subsequent experiments focus on comparing the performance of the original and improved HO in path planning and further evaluate the proposed framework in structured library environments to assess its practical navigation performance in real library scenarios.

(2)Path Refinement Performance in Structured Library Layouts

To evaluate the effectiveness of the proposed refinement strategy, representative bookshelf and rest-area layouts with different structural configurations were constructed in MATLAB, and simulation experiments were conducted to compare the improved Hippopotamus Optimization (HO) algorithm with the original HO. The path-planning and iteration results are presented in **Fig. 5**, where the red and green lines represent the original and improved HO, respectively, and the black and white squares denote obstacles and traversable areas.

ISSN: 2688-8645

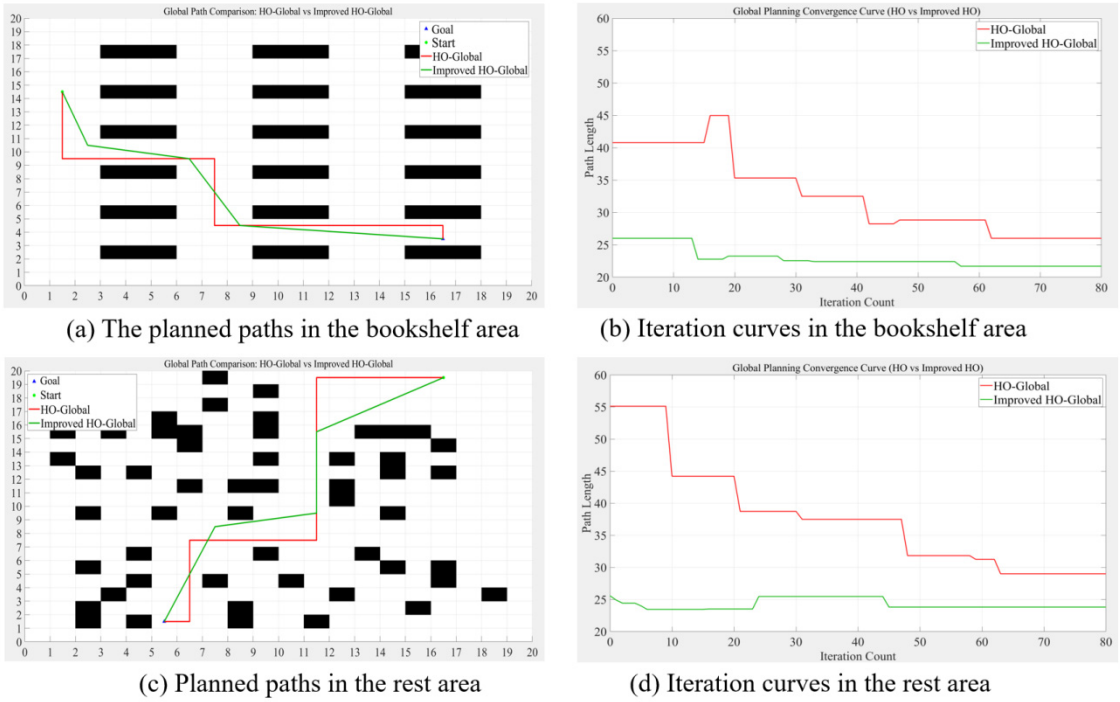
Table 2(a) Benchmark comparison results on selected CEC2005 test functions

Function	SSA (Mean±Std)	HO (Mean±Std)	Improved HO (Mean±Std)
F1	3.685E-19±2.016E-18	0 ± 0	0 ± 0
F2	1.820E-09±1.041E-08	2.633E-184±1.542E-183	5.093E-186±3.589E-185
F3	5.180E-15±2.072E-14	0 ± 0	0 ± 0
F4	9.469E-11±5.757E-10	2.564E-183±1.340E-182	3.457E-190±2.050E-189
F8	−6924.4±824.06	−9490.3±1294.6	−11432±1781.6
F15	3.182E-04±7.325E-05	3.075E-04±1.111E-07	3.074E-04±9.181E-09

Table 2(b) Non-parametric statistical test results on selected CEC2005 benchmark functions

Function	Kruskal-Wallis p-value	Improved HO vs. HO (Holm-adjusted p-value)	Improved HO vs. SSA (Holm-adjusted p-value)
F1	1.516E-30	1.000	1.511E-18
F2	4.732E-23	9.393E-09	4.266E-15
F3	6.529E-29	1.000	1.860E-17
F4	4.619E-25	7.207E-11	1.349E-15
F8	4.246E-23	4.479E-08	5.240E-15
F15	2.686E-19	0.0381	2.160E-15

Note: A Holm-adjusted p-value < 0.05 indicates statistical significance.



(a) The planned paths in the bookshelf area (b) Iteration curves in the bookshelf area

(c) Planned paths in the rest area (d) Iteration curves in the rest area

Fig 5. Simulation comparison of the original and improved HO in structured library layouts.

In the bookshelf area, the improved HO reduced the path length from 26 to 21.94 m, the number of bend points from 4 to 3, the run time from 1.901 to 1.305 s, and the iteration count from 62 to 57. In the rest area, these values decreased from 29 to 23.8 m, 4 to 3 bend points, 2.203 to 1.565 s, and 63 to 45 iterations, respectively. The simulation results demonstrate that the improved HO generates shorter paths with fewer bend points while reducing run time and iteration count in both structured layouts.

3.3. Coordination of Deterministic Search and Heuristic Optimization

In the proposed global planning strategy, deterministic search and heuristic optimization are explicitly coordinated within a unified decision-making pipeline rather than independently applied. The improved A* algorithm first provides structure-aware global guidance by

ISSN: 2688-8645

generating a feasible path under static layout constraints, which serves as structured prior information for subsequent refinement.

Based on this prior knowledge, the improved Hippopotamus Optimization (HO) algorithm performs heuristic refinement on selected control points, improving node distribution and overall path smoothness while preserving global feasibility. Through this coordinated mechanism, the global planner effectively balances deterministic feasibility and heuristic exploration under complex layout constraints. The execution flow of the proposed coordination strategy is illustrated in **Fig. 6.**

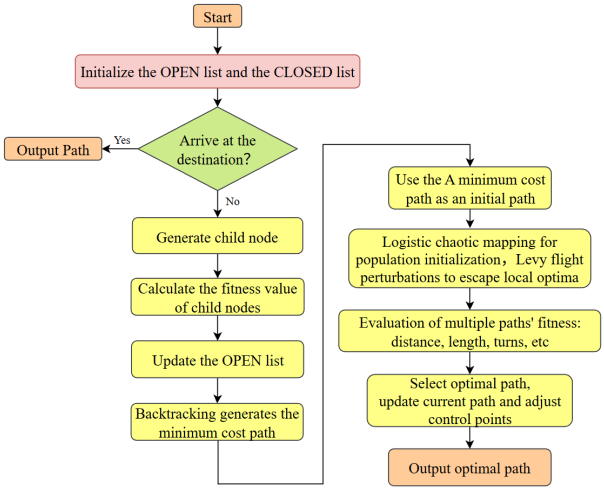
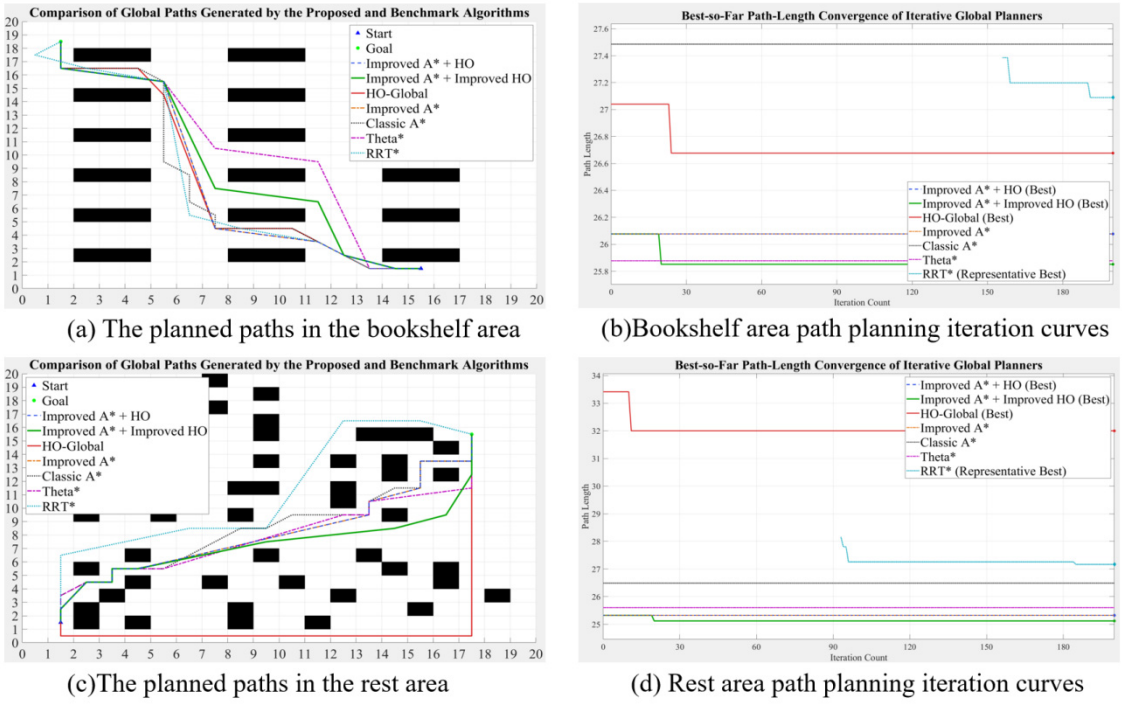


Fig 6. Execution flow of the coordinated global decision-making strategy integrating deterministic search and heuristic optimization.



(a) The planned paths in the bookshelf area (b)Bookshelf area path planning iteration curves

(c)The planned paths in the rest area (d) Rest area path planning iteration curves

Fig 7. Simulation comparison of the proposed and benchmark path planning approaches in structured library layouts.

To evaluate the effectiveness of the proposed coordination strategy, comparative simulation experiments were conducted in representative bookshelf and rest-area scenarios. The planned paths and iteration curves are presented in **Fig. 7**, while their quantitative performance is summarized in **Table 3(a)** and **(b)**.

ISSN: 2688-8645

Table 3(a) Performance comparison of seven path planning algorithms for bookshelf area

Evaluation Metrics / Algorithm	Path length (m)	Number of bend points (count)	Run time (s)	Iteration count (times)	Number of Expanded Nodes (count)
Improved A*+ Classic HO	26.077	6	2.5093	200	—
Improved A*+Improved HO	25.852	6	2.1605	20	—
Classic HO	26.677	1	2.4381	24	—
Improved A*	26.077	6	0.0262	—	156
Classic A*	27.485	10	0.0672	—	198
Theta*	25.878	5	0.0725	—	141
RRT*	27.089	7	0.2362	—	191

Table 3(b) Performance comparison of seven path planning algorithms for rest area

Evaluation Metrics / Algorithm	Path length (m)	Number of bend points (count)	Run time (s)	Iteration count (times)	Number of Expanded Nodes (count)
Improved A*+ Classic HO	25.321	10	2.8389	200	—
Improved A*+Improved HO	25.119	9	2.1030	20	—
Classic HO	32.000	2	1.5060	11	—
Improved A*	25.321	10	0.0271	—	149
Classic A*	26.485	14	0.0504	—	182
Theta*	25.600	9	0.0686	—	161
RRT*	27.165	5	0.7637	—	185

Note: Iteration count refers to the HO optimization stage, whereas expanded nodes are reported for standalone graph-search and sampling-based planners; these metrics are not directly comparable. "—" indicates an inapplicable or unreported metric.

The results in **Table 3(a)** and **(b)** reveal different optimization behaviors under the two structured layouts. In the bookshelf area, dense static obstacles and narrow passages increase the risk of local entrapment, causing the original HO to converge to a less favorable path. In contrast, the proposed hybrid strategy maintains search diversity during refinement, reduces the influence of local optima, and achieves more efficient convergence while preserving global path quality. In the rest area, the proposed method also exhibits more stable optimization behavior than Improved A* + Classic HO, reducing unnecessary detours and improving path quality.

3.4. Geometric Path Smoothing for Global Plan Feasibility

Although the refined global path achieves improved optimality, it may still contain sharp turns that limit execution feasibility for real robots. Therefore, geometric path smoothing is introduced to ensure continuity and curvature consistency before local planning. Catmull-Rom spline interpolation is adopted because it preserves the original path nodes while ensuring C^1continuity, enabling smooth trajectory generation without introducing additional control points that may distort the global path topology. Its geometric matrix representation is:

$$P(t)=\begin{bmatrix}1 & t & t^2 & t^3\end{bmatrix}\begin{bmatrix}0 & 1 & 0 & 0\\ -\tau & 0 & \tau & 0\\ 2\tau & \tau-3 & 3-2\tau & -\tau\\ -\tau & 2-\tau & \tau-2 & \tau\end{bmatrix}\begin{bmatrix}P_{i-2}\\ P_{i-1}\\ P_i\\ P_{i+1}\end{bmatrix} \tag{18}$$

ISSN: 2688-8645

Where $P(t)$ denotes the point coordinates on the curve at parameter t (for a 2D path, it is (x, y)). P_{i-2}, P_{i-1}, P_i, P_{i+1} are four consecutive control points. The curve segment precisely passes through P_{i-1} and P_i. The interpolation parameter t is defined over the continuous interval [0, 1], and each value of t maps uniquely to a point on the curve. When $t=0$, the point is at P_{i-1}; when $t=1$, it is at P_i. The parameter τ controls the curve's tension. Using τ = 0.5 (known as centripetal parameterization) generally produces well-behaved curves without loops or sharp corners. The path smoothing results for different library layouts are presented in the figures below.

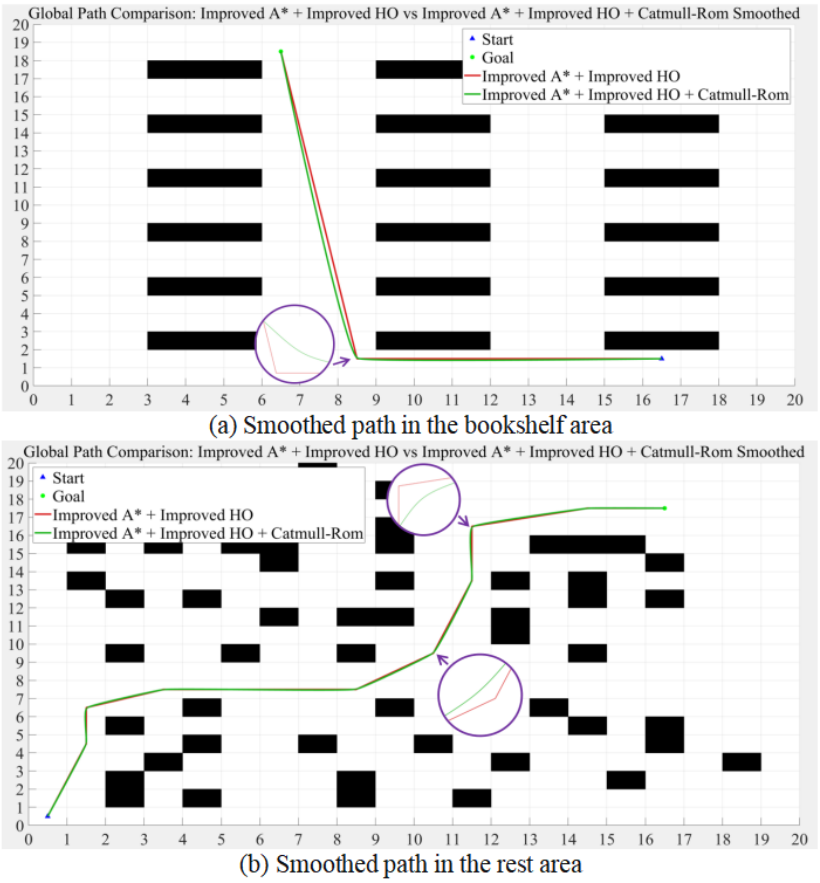


Fig 8. Path smoothing results in structured library layouts

4. Local Reactive Planning Under Dynamic Constraints

To address the limited real-time responsiveness of global planning to dynamic obstacles and local uncertainties, a local reactive planner generates short-horizon, kinematically feasible trajectories under global path guidance.

In this study, a Timed Elastic Band (TEB)-based planner is adopted for real-time trajectory adjustment, as it explicitly incorporates kinematic and temporal constraints, enabling smooth and dynamically feasible trajectory generation while maintaining consistency with the global path. The local planning problem is formulated as a nonlinear optimization problem, expressed as follows:

Optimization variables $B=(Q,T)=\{q_0,q_1,\cdots q_n,\Delta t_0,\Delta t_1,\cdots \Delta t_{n-1}\}$.

Wherein, Q is termed the pose sequence, $Q=\{q_0,q_1\cdots q_n\}$, $q_i=(x_i,y_i,\theta_i)$ .T is termed the time interval sequence, $T=\{\Delta t_0,\Delta t_1\cdots \Delta_{n-1}\}$, $\Delta t_i=t_{i+1}-t_i>0$.

Weighted multi-objective optimization,

ISSN: 2688-8645

$$\min_B F(B) = \sum_k^5 \gamma_k \cdot f_k(B) = \gamma_{time} \cdot f_t + \gamma_{obstacle} \cdot f_{obstacle} + \gamma_{path} \cdot f_{path} + \gamma_{kin} \cdot f_{kin} + \gamma_{smooth} \cdot f_{smooth} \tag{19}$$

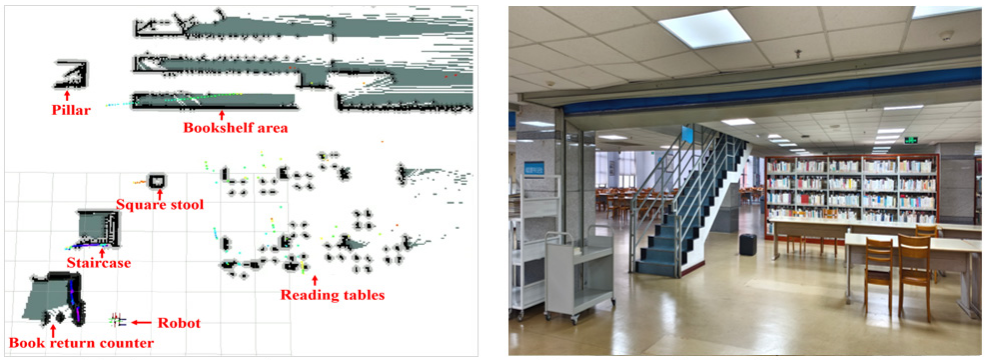
$$s.t. \begin{cases} q_0 = q_{start}, \ q_n = q_{goal} \\ v_i \le v_{\max}, \ |\omega_i| \le \omega_{\max} \end{cases}$$

$$\begin{cases} f_t = \sum_{i=0}^{n-1} (\Delta t_i)^2 \\ f_{obstacle} = \sum_{i=0}^{n} \Phi[d(q_i)], \Phi[d(q_i)] = \begin{cases} \infty, & d \le d_{\min} \\ k_{obstacle}(d - d_{\min})^{-2}, & d_{\min} < d < d_{safe} \\ 0, & d \ge d_{safe} \end{cases} \\ f_{path} = \sum_{i=0}^{n} \left\| project(q_i) - w_i^* \right\|^2 \\ f_{kin} = \sum_{i=0}^{n-2} \left[\gamma_{v_x} (\Delta v_{x,i})^2 + \gamma_{v_y} (\Delta v_{y,i})^2 + \gamma_{\omega} (\Delta \omega_i)^2 \right] \\ f_{smooth} = \sum_{i=1}^{n-1} \left\| q_{i-1} - 2q_i + q_{i+1} \right\|_2^2 \end{cases}$$

Where f_t is the minimum time term, $f_{obstacle}$ is the obstacle avoidance (safety) term, f_{path} is the path following term, f_{kin} is the kinodynamic smoothing term, f_{smooth} is the pose smoothing term. $\gamma_{time} = 1$, $\gamma_{obstacle} \in (50,200)$, $\gamma_{path} \in (0.5,2)$, $\gamma_{kin_v_x} = \gamma_{kin_v_y} = 10$, $\gamma_{kin_\omega} = 5$, $\gamma_{smooth} = 2$. The obstacle avoidance penalty coefficient $k_{obstacle}$ usually takes a value between 10 and 500. $project(q_i)$ denotes the nearest point where pose $q_i = (x_i, y_i)$ is projected onto the global reference path. $w_i^* = (x_i^*, y_i^*)$ denotes the waypoint on the global reference path nearest to pose q_i.

5. Experimental Evaluation

5.1. Experiment Setup



(a) The constructed SLAM map (b) Corresponding real library environment

Fig 9. SLAM map and corresponding real library environment.

Experimental evaluation was conducted on a real mobile robot platform under a ROS-based navigation framework. A LiDAR sensor was used for environment perception and map construction, and the proposed global–local planning framework was integrated into the navigation stack. The constructed SLAM map and corresponding real library environment are shown in **Fig. 9.**

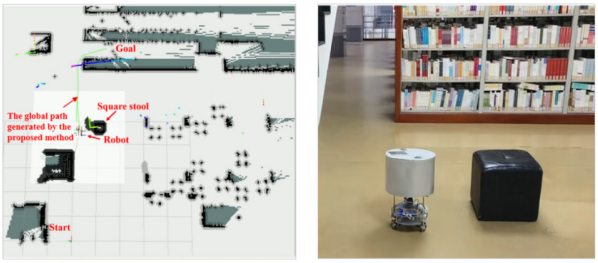
ISSN: 2688-8645

5.2. Navigation Performance Evaluation in Structured Indoor Environments

To evaluate the robustness and adaptability of the proposed framework during navigation, real-world experiments were conducted in a library room environment featuring representative structured constraints. The test scenario included a square stool as a static obstacle, a pedestrian as a dynamic obstacle, and a U-shaped bookshelf aisle.

(1) Static obstacle

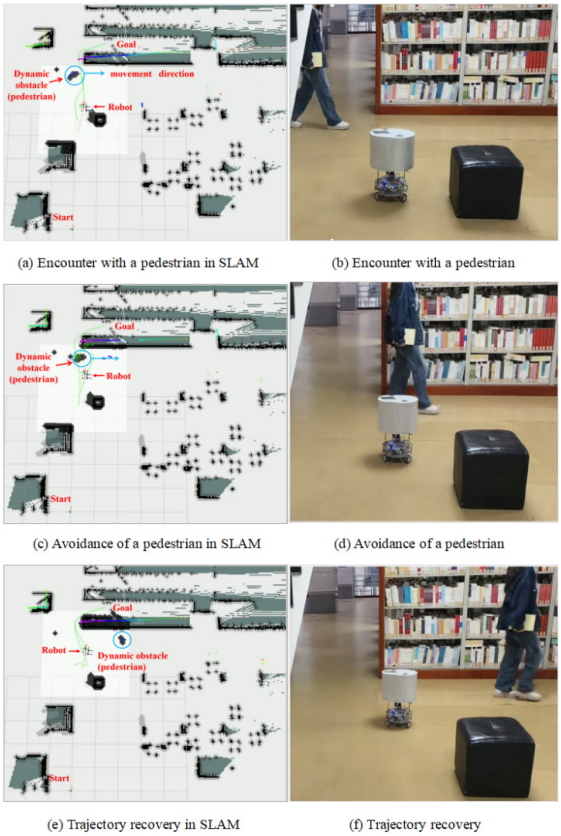
The static obstacle avoidance experiment is presented in **Fig. 10**, including the SLAM-based navigation result and the corresponding real-world result.



(a) Static obstacle avoidance in SLAM (b) Static obstacle avoidance result

Fig 10. Static obstacle avoidance experiment

The results show that the proposed framework effectively adjusts the local trajectory to avoid the static obstacle while maintaining consistency with the global path.



(a) Encounter with a pedestrian in SLAM (b) Encounter with a pedestrian

(c) Avoidance of a pedestrian in SLAM (d) Avoidance of a pedestrian

(e) Trajectory recovery in SLAM (f) Trajectory recovery

Fig 11. Dynamic pedestrian obstacle avoidance and trajectory recovery in the library environment.

(2) Dynamic obstacle avoidance test

The results of dynamic obstacle avoidance experiment are shown in **Fig. 11.**

Dynamic obstacle avoidance experiments further demonstrate strong local reactivity under human-induced disturbances.

ISSN: 2688-8645

(3) U-Shaped aisle

The experiment results are shown in **Fig. 12.**

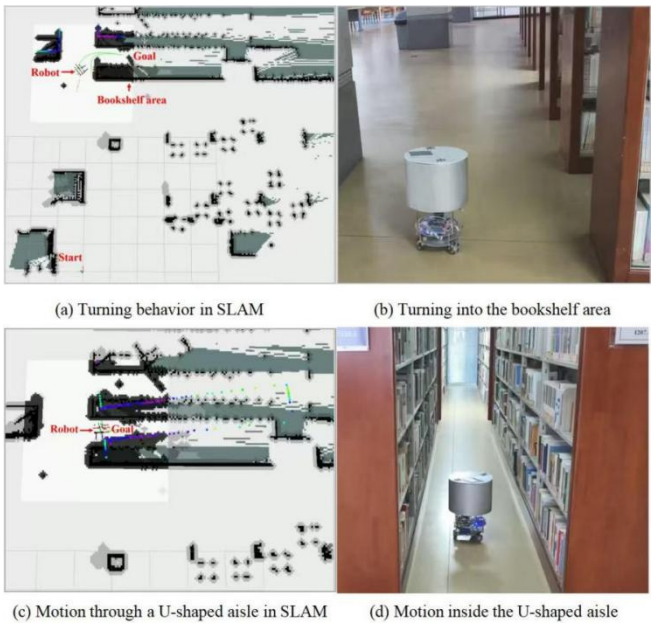


Fig 12. U-shaped aisle navigation experiment in the library environment.

The U-shaped aisle navigation test validates the framework's capability to handle constrained passage structures commonly encountered in structured indoor environments.

6. Conclusion

The following conclusions are drawn from the theoretical analysis, simulations, and experiments conducted in this study:

(1) This study proposes an integrated global–local path planning framework for structured indoor environments. By coordinating deterministic global search, heuristic optimization, and local reactive planning within a unified decision-making pipeline, the proposed framework effectively addresses navigation challenges arising from dense static obstacles, narrow passage structures, and dynamic disturbances.

(2) The simulation results indicate that the improved HO achieved the best or jointly best mean performance on all six benchmark functions. The non-parametric statistical analysis further demonstrated statistically significant improvements over SSA on all six functions, thereby confirming the effectiveness of the proposed improvement strategies. However, slightly larger standard deviations were occasionally observed, indicating that the algorithm's stability could be further improved.

(3) Experimental results from simulations and real-world library scenarios demonstrate that the proposed approach achieves improved global path quality and enhanced local adaptability compared with conventional planning methods. Future work will focus on extending the framework to more complex indoor environments and further strengthening knowledge-guided decision-making for adaptive robotic navigation.

ISSN: 2688-8645

Declarations

Conflicts of Interest

All authors declare that they have no conflicts of interest.

Informed Consent

Written informed consent was obtained from the participant involved in the dynamic pedestrian obstacle-avoidance experiment and for the publication of the associated images.

References

[1] Zhu, D. D., & Sun, J. Q. (2021). A new algorithm based on Dijkstra for vehicle path planning considering intersection attributes. IEEE Access, 9, 19761–19775. https://doi.org/10.1109/ACCESS.2021.3053712.

[2] Zhang, F., Xia, R., & Chen, X. (2020). An optimal trajectory planning algorithm for autonomous trucks: Architecture, algorithm, and experiment. International Journal of Advanced Robotic Systems, 17(2). https://doi.org/10.1177/1729881420909603.

[3] Zhang, D., Chen, C., & Zhang, G. (2024). AGV path planning based on improved A-star algorithm. In 2024 IEEE 7th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC) (pp. 1–6). https://doi.org/10.1109/IAEAC60237.2024.10501420.

[4] Tang, X. R., Zhu, Y. K., & Jiang, X. X. (2021). Improved A-star algorithm for robot path planning in static environment. Journal of Physics: Conference Series, 1792(1), 012045. https://doi.org/10.1088/1742-6596/1792/1/012045.

[5] Li, X., & Tong, Y. (2023). Path planning of a mobile robot based on the improved RRT algorithm. Applied Sciences, 14(1), 25. https://doi.org/10.3390/app14010025.

[6] Zhang, L., et al. (2023). Mobile robot path planning algorithm based on RRT-Connect. Electronics, 12(11), 2456. https://doi.org/10.3390/electronics12112456.

[7] Chen, J., Zhao, Y., & Xu, X. (2021). Improved RRT-Connect based path planning algorithm for mobile robots. IEEE Access, 9, 145988–145999. https://doi.org/10.1109/ACCESS.2021.3122783.

[8] Amiri, M. H., Mehrabi, H. N., Montazeri, M., Mirjalili, S., & Khodadadi, N. (2024). Hippopotamus optimization algorithm: A novel nature-inspired optimization algorithm. Scientific Reports, 14, 5032. https://doi.org/10.1038/s41598-024-55561-4.

[9] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

[10] Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., & Mirjalili, S. M. (2017). Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in Engineering Software, 114, 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002.

[11] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of ICNN'95—International Conference on Neural Networks (Vol. 4, pp. 1942–1948). https://doi.org/10.1109/ICNN.1995.488968.

[12] Song, J., Hao, C., & Su, J. (2020). Path planning for unmanned surface vehicle based on predictive artificial potential field. International Journal of Advanced Robotic Systems, 17(2). https://doi.org/10.1177/1729881420918461.

[13] Liu, L., Wang, B., & Xu, H. (2022). Research on path-planning algorithm integrating optimized A-star algorithm and artificial potential field method. Electronics, 11(22), 3660. https://doi.org/10.3390/electronics11223660.

[14] Xu, W., et al. (2023). A local path planning algorithm based on improved dynamic window approach. Journal of Intelligent & Fuzzy Systems, 45(3), 4917–4933. https://doi.org/10.3233/JIFS-230274.

ISSN: 2688-8645

[15] Lai, X., et al. (2023). Enhanced DWA algorithm for local path planning of mobile robot. Industrial Robot: The International Journal of Robotics Research and Application, 50(1), 186–194. https://doi.org/10.1108/IR-04-2022-0085.

[16] Alawida, M. (2024). Enhancing logistic chaotic map for improved cryptographic security in random number generation. Journal of Information Security and Applications, 80, 103685. https://doi.org/10.1016/j.jisa.2024.103685.