Robotica (2025), **43**, pp. 1929–1952
doi:10.1017/S026357472500061X

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Path planning for mobile robots in complex environments based on enhanced sparrow search algorithm and dynamic window approach

Yixuan Luo, Shusen Lin, Yifan Wang and Kai Liang

College of Intelligent Manufacturing, Taizhou University, Taizhou, 318001, China
Corresponding author: Shusen Lin; Email: linshusen@tzc.edu.cn

Received: 1 January 2025; **Revised:** 25 March 2025; **Accepted:** 2 April 2025; **First published online:** 16 May 2025

Keywords: mobile robot; sparrow search algorithm; dynamic window approach; path planning; dynamic obstacle avoidance

Abstract

Traditional path planning algorithms often encounter challenges in complex dynamic environments, including local optima, excessive path lengths, and inadequate dynamic obstacle avoidance. Thus, the development of innovative path planning algorithms is essential. This article addresses the challenges of mobile robot path planning in complex environments, where traditional methods often converge to local optima, leading to suboptimal path lengths, and struggle with dynamic obstacle avoidance. To overcome these limitations, we propose an integrated algorithm, the enhanced sparrow search algorithm combined with the dynamic window approach (ESSA-DWA). The algorithm first utilizes ESSA for global path planning, followed by local path planning facilitated by the DWA. Specifically, ESSA incorporates Tent chaotic initialization to enhance population diversity, effectively mitigating the risk of premature convergence to local optima. Moreover, dynamic adjustments to the inertia weight during the search process enable an adaptive balance between exploration and exploitation. The integration of a local search strategy further refines individual updates, thereby improving local search performance. To enhance path smoothness, the Floyd algorithm is employed for path optimization, ensuring a more continuous trajectory. Finally, the combination of ESSA and DWA uses key nodes from the global path generated by ESSA as reference points for the local planning process of DWA. This approach ensures that the local path closely follows the global path while also enabling real-time dynamic obstacle detection and avoidance. The effectiveness of the algorithm has been validated through both simulations and practical experiments, offering an efficient and viable solution to the path planning problem.

1. Introduction

A mobile robot is a mechanical system designed for autonomous or semiautonomous movement across diverse environments. It must be equipped with highly sensitive sensors to perceive its surroundings, accurately determine its position, and construct detailed environmental maps, thereby enabling safe navigation to designated destinations and the completion of predefined tasks [1]. In the context of path planning, methods are generally categorized into global and local path planning, depending on the extent of environmental knowledge available. Global path planning assumes that the robot has comprehensive information about the environment, allowing it to follow a predetermined path to reach its destination [2]. Traditional global path planning algorithms include Dijkstra's algorithm [3], the A* algorithm [4], rapidly exploring random trees (RRTs) [5], and the probabilistic roadmap (PRM) [6]. In contrast, local path planning is employed when the robot has limited or no prior knowledge of the environment, relying on real-time sensing and dynamic decision-making. Traditional local path planning algorithms include the artificial potential field (APF) method [7] and the dynamic window approach (DWA) [8].

Significant advancements have been made in the field of path planning. Savkin et al. [9] proposed a reactive randomized algorithm that ensures the robot avoids obstacles with probability 1 and reaches

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

the specified goal; however, the planned path is not guaranteed to be the shortest, and the handling of dynamic obstacles is not addressed. Zhou et al. [10] applied an improved Dijkstra algorithm to complex product integration scheduling, enhancing path connectivity; however, this approach introduced relatively high computational complexity. Jabbar et al. [11] addressed the shortest path problem by enhancing the Dijkstra algorithm with a reverse priority queue, improving computational efficiency. Tang et al. [12] proposed a geometric A* algorithm tailored for port environments, optimizing paths using a screening function and interpolation technique, which reduced the number of turns and shortened the path length. Li et al. [13] combined an improved A* algorithm with the DWA, significantly improving dynamic obstacle avoidance, reducing both path length and planning time, and enhancing the robot's speed. Additionally, Dang et al. [14] proposed a hybrid path planning method that integrates an enhanced A*, Jump Point Search, and the improved DWA, while further augmenting the IRDC-Net model to achieve better navigation in dynamic environments with moving obstacles. Liang et al. [15] developed the CCPF-RRT* algorithm for crowded environments, incorporating a mobility cost function based on congestion intensity and path length to obtain an optimal path with minimal cost. Novosad et al. [16] introduced the CTopPRM algorithm, which can identify multiple topologically distinct paths in complex 3D environments, increasing the likelihood of exploring all potential routes. In the realm of local path planning, Zhang et al. [17] addressed the speed issue for Autonomous Underwater Vehicles (AUVs) in complex obstacle-laden environments by combining the A* and APF algorithms for fast and effective obstacle avoidance. Li et al. [18] proposed the APF-ant colony optimization (ACO) algorithm, enhancing collision avoidance and trajectory planning for ships in dynamic environments by optimizing the APF method and integrating it with the ACO algorithm. Pan et al. [19] designed the D2WA algorithm, which dynamically incorporates obstacle velocity information to improve the navigation safety of Automated Guided Vehicles (AGVs).

In recent years, heuristic optimization algorithms have garnered significant attention in the field of path planning, becoming a focal point for scholarly research. Several classical heuristic optimization algorithms have been widely applied, including the genetic algorithm (GA) [20], particle swarm optimization (PSO) [21], ACO [22], Grey Wolf optimization (GWO) [23], Whale optimization algorithm (WOA) [24], and pigeon-inspired optimization (PIO) [25]. As research has advanced, efforts have been made to refine and optimize these algorithms to improve their performance in path planning, yielding promising results. For example, Li et al. [26] proposed an improved algorithm that combines GA with the DWA, significantly enhancing pick-up efficiency in logistics systems and generating smoother paths with fewer turns. Lin et al. [27] developed a hybrid PSO-SA algorithm aimed at minimizing the path length of AGVs, while ensuring collision-free and smooth paths. Zhou et al. [28] introduced an initialization method with unevenly distributed pheromones for ACO to improve its search capabilities in the initial path planning stage. By incorporating adaptive pheromone evaporation factors and an update mechanism, the algorithm was able to generate smoother and shorter paths. Furthermore, Yu et al. [29] combined GWO with differential evolution, resulting in enhanced optimization performance for path planning. To address challenges faced by indoor mobile robots in obstacle avoidance, Dai et al. [30] proposed an improved WOA algorithm that incorporates adaptive techniques, an enhanced potential field factor, and a virtual obstacle mechanism, improving convergence speed and avoiding local optima. Additionally, Jiang et al. [31] successfully integrated an improved fuzzy clustering algorithm with PIO for fixed-wing UAV trajectory planning, achieving a more balanced clustering effect and enhancing the comprehensiveness and accuracy of trajectory exploration.

The sparrow search algorithm (SSA), introduced by Xue [32] in 2020, is a novel metaheuristic optimization algorithm inspired by the foraging behavior of sparrows. Due to its exceptional global search capabilities and strong optimization performance, SSA has garnered significant attention from researchers. Zheng et al. [33] improved SSA to optimize a backpropagation (BP) neural network for inland vessel trajectory prediction, resulting in significant improvements in prediction accuracy and stability. Khedr et al. [34] proposed a path planning method for mobile receivers in wireless sensor networks based on an enhanced SSA, which notably improved data collection efficiency and network robustness in dynamic environments. Liu et al. [35] applied SSA to unmanned ship path planning, incorporating

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

cubic chaotic mapping and Gaussian random walk strategies to generate stable and reliable navigation paths. Wang et al. [36] implemented PSO in parallel with an Enhanced SSA, achieving more feasible and effective path planning outcomes. Liu et al. [37] further Improved SSA by integrating it with the DWA to develop the ISSA-DWA algorithm for optimal path planning in mobile robots. This approach demonstrated faster convergence and higher path quality, though it has yet to see widespread validation in practical applications. Hou et al. [38] proposed a multistrategy enhanced SSA integrated with the dynamic window approach (MISSA-DWA), improving both the efficiency and safety of robot motion. However, the generated path still exhibits limitations in smoothness.

This article tackles the challenges faced by mobile robots in complex environments, where the SSA often falls into local optima, produces excessively long paths, and struggles with dynamic obstacle avoidance. Building upon the existing body of research, this article proposes a hybrid algorithm that combines the enhanced sparrow search algorithm (ESSA) with the DWA. ESSA incorporates several advanced techniques, including Tent chaotic initialization, dynamic inertia weighting, a local search strategy, and the Floyd path optimization algorithm, to enhance both the global search capability and convergence speed of the algorithm. First, ESSA employs Tent chaotic initialization to increase population diversity, effectively mitigating the risk of premature convergence to local optima. Second, dynamic inertia weighting is incorporated, adapting during the search process to strike an optimal balance between exploration and exploitation. Additionally, the inclusion of a local search strategy refines individual updates, improving the precision of the search and enhancing the algorithm's local search ability. Furthermore, the Floyd path optimization algorithm is applied to optimize the generated path, making it smoother and enhancing both the practical applicability and accuracy of the algorithm. Finally, the ESSA and DWA algorithms are integrated by extracting key nodes from the global path planning of the ESSA algorithm, which are then used as references for the local path planning in the DWA algorithm. This integration ensures that the local path is more aligned with the global path, enabling dynamic obstacle avoidance. The algorithm's effectiveness and practicality have been confirmed through both simulation and practical experiments, offering an efficient and viable solution to path planning problems.

The main contributions of this study are as follows:

- The ESSA is proposed, which improves the search performance and stability of the SSA algorithm by incorporating Tent chaotic mapping, dynamic inertia weights, and local search strategies. The Floyd algorithm is then employed to optimize the path, resulting in a smooth, optimal trajectory.
- The ESSA and DWA algorithms are integrated by extracting key nodes from the global path planning phase and using them as reference points for local path planning in DWA. This integration allows for dynamic obstacle detection and the generation of local avoidance paths, which better meet practical requirements. The effectiveness and superiority of the integrated algorithm are validated through simulation experiments in complex environments.
- The ESSA-DWA fusion algorithm is applied to mobile robots, and its effectiveness is verified through both dynamic and static experiments, providing an efficient and viable solution to path planning challenges.

The remainder of this article is organized as follows: Section 2 introduces the standard SSA. Section 3 details the SSA with multiple strategic improvements. Section 4 provides an overview of DWA and explains the principles behind the fusion of the two algorithms. Section 5 presents simulation results for global path planning, the integrated fusion algorithm, and practical applications of the hybrid algorithm in both dynamic and static environments with mobile robots. Finally, Section 6 concludes the study and outlines directions for future research.

2. Sparrow search algorithm

In the SSA, sparrows are categorized into three roles based on their foraging and antipredation behaviors: producers, scroungers, and vigilantes. Each sparrow's position in the search space corresponds to a

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

potential solution. Producers identify optimal foraging areas for the group, scroungers follow producers to obtain food, and vigilantes monitor for predators and alert the group. Sparrows can switch roles, though the overall distribution of roles remains constant. When the vigilance level exceeds a predefined threshold, the group enters antipredation mode and flies to a safe zone.

The position of the sparrow swarm in the search space is represented by

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,d} \end{pmatrix} \tag{1}$$

where n is the number of sparrows and d is the dimension of the optimization variable space. The fitness of all sparrows is given by

$$F_X = \begin{bmatrix} f([X_{1,1} & \cdots & X_{1,d}]) \\ f([X_{2,1} & \cdots & X_{2,d}]) \\ & \vdots & \\ f([X_{n,1} & \cdots & X_{n,d}]) \end{bmatrix} \tag{2}$$

where each function f_x in the row represents the fitness of each sparrow. Producers with better fitness lead the group and have priority access to food. Their position update formula is

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(\frac{-i}{\alpha \cdot MaxIter}\right), & \text{if } R_2 < ST \\ X_{i,j}^{t} + Q \cdot L, & \text{if } R_2 \geq ST \end{cases} \tag{3}$$

where t is the current iteration, $X_{i,j}^{t+1}$ is the position information of the i-th sparrow in the j-dimension at the $(t+1)$-th iteration, $MaxIter$ is the maximum number of iterations, $\alpha \in (0, 1)$ is a random number, $R_2 \in [0, 1]$ is the warning value, $ST \in [0.5, 1]$ is the safe value, Q is the random number that obeys the normal distribution, L is the matrix with a size of $1 \times d$ and all elements are 1, and d is the dimension. If $R_2 < ST$, there are no predators, and the producer can explore for better food sources, if $R_2 > ST$, the group must perform antipredation behavior.

Scroungers update their positions based on the producers' best positions and the worst position in the current population:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^{t} - X_{i,j}^{t}}{i^2}\right), & \text{if } i > \frac{n}{2} \\ X_P^{t+1} + \left|X_{i,j} - X_P^{t+1}\right| \cdot A^T (AA^T)^{-1} \cdot L, & \text{if } i \leq \frac{n}{2} \end{cases} \tag{4}$$

where t indicates the current iteration, X_P^t represents the current best position of the producers, X_{worst}^t represents the worst position in the current iteration population, A is a $1 \times d$ matrix where each element is randomly chosen as ± 1, n represents the population size, and L is a $1 \times d$ matrix where all elements are 1. If $i > n/2$, the scrounger searches for new food sources; if $i < n/2$, the scrounger follows the producers.

The algorithm assumes that $10\% \sim 20\%$ of the sparrows can sense danger. Vigilantes update their positions based on the global best and worst positions, as well as their own fitness:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta \cdot \left|X_{i,j}^{t} - X_{best}^{t}\right|, & \text{if } f_i > f_g \\ X_{i,j}^{t} + K \cdot \left(\frac{\left|X_{i,j}^{t} - X_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right), & \text{if } f_i = f_g \end{cases} \tag{5}$$

where X_{best}^t represents the best position during the iteration, β is a step size adjustment parameter that follows a normal distribution with a mean of 0 and a variance of 1, $K \in [-1, 1]$ is a random number, and ε is a small constant to avoid division by zero. f_i, f_g, and f_w are the fitness values of the current, global best, and worst individuals, respectively. The vigilante adjusts its position based on whether its fitness is higher or equal to the global best.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

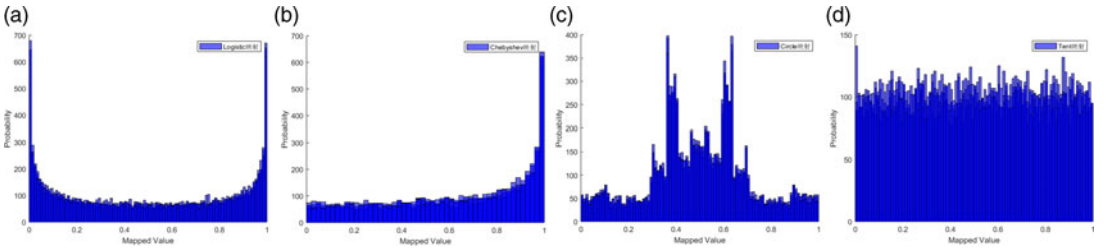


Figure 1. *Frequency distribution histograms of chaotic maps.*

3. Global path planning with ESSA

3.1. Tent map chaotic initialization

When initializing the sparrow population, random functions are commonly used to generate positions, but this often results in uneven distribution, limiting the global search capability in the early optimization stages. Chaotic maps, nonlinear dynamical systems that appear random but are deterministically generated, can address this issue. These maps exhibit properties such as randomness, ergodicity, sensitivity to initial conditions, and unpredictability. The Logistic, Chebyshev, Circle, and Tent mappings are each iterated 10,000 times to generate distribution frequency histograms within the [0, 1] range, as shown in Figure 1.

Figure 1 illustrates the probability distribution characteristics of four chaotic maps within the solution space in the form of histograms. The Logistic map exhibits a pronounced bimodal distribution, with significant spikes occurring at the boundaries. The Chebyshev map demonstrates a unimodal distribution, characterized by a prominent peak near the value of 1. The circle map displays a multipeak and irregular distribution profile. These features may lead to nonuniform exploration of the global search space, requiring sparrow individuals multiple iterations to locate the optimal solution. By comparison, the Tent map presents a more uniform frequency distribution, offering superior ergodicity and faster convergence speed. This characteristic facilitates more effective utilization of the search space and enhances the quality of the algorithm's initial solutions.

The Tent map system is expressed as

$$x_{n+1} = \begin{cases} r \cdot x_n, & \text{if } x_n < 0.5 \\ r \cdot (1 - x_n), & \text{if } x_n \geq 0.5 \end{cases} \tag{6}$$

where x_{n+1} represents the sparrow individual after mapping iteration, and r is the control parameter, typically valued within the interval [0, 1]. The sparrow population generated using the Tent map exhibits more uniform distribution and greater diversity, reducing edge aggregation and enhancing the algorithm's global search capability, thus mitigating premature convergence.

3.2. Dynamic inertia weight

In traditional SSA, rapid convergence toward the global optimum early in the process reduces global search capabilities and increases the risk of getting trapped in local optima. To address this, we introduce a dynamic inertia weight to update the positions of the producers in the sparrow population. The inertia weight is increased during early iterations to promote global exploration and gradually decreased in later stages to focus on local search around the best solutions. This approach enhances the algorithm's local search ability and mitigates premature convergence to local optima.

The improved position update method for producers using inertia weight is shown in Eq. (7):

$$X_{i,j}^{t+1} = \begin{cases} \omega_t \cdot X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot MaxIter}\right), & \text{if } R_2 < ST \\ \omega_t \cdot X_{i,j}^t + Q \cdot L, & \text{if } R_2 \geq ST \end{cases} \tag{7}$$

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

where the dynamic inertia weight ω_t is defined as in Eq. (8):

$$\omega_t = \omega_1 - \frac{\omega_{1_}\omega_2}{MaxIter} \cdot t \tag{8}$$

where ω_1 and ω_2 are weight adjustment parameters, which are set to $\omega_1 = 0.9$ and $\omega_2 = 0.4$ in this study. Other parameters are detailed in Eq. (3) and will not be elaborated further.

3.3. Local search strategy

This study introduces a local search strategy based on the current global optimal solution to enhance its quality. The strategy performs a multidimensional search at each node of the global optimal individual during the current iteration. It systematically explores each dimension, evaluating the results using a fitness function. These results are then compared with the presearch fitness values to determine if an improved solution is found. This approach enhances the algorithm's global search capability and overall optimization performance. The process is described by Eq. (9).

$$X^t_{best,i} = \begin{cases} R_{best,i}, & \text{if } f_R < f_b \\ X^t_{best,i}, & \text{if } f_R \geq f_b \end{cases} \tag{9}$$

where $X^t_{best,i}$ represents the i-th node of the global optimal individual at iteration t, and $R_{best,i}$ denotes the new node found based on the i-th node, which is a random integer within the dimension d. f_R is the new fitness value obtained by searching $R_{best,i}$, and f_b is the fitness value of the original optimal individual. i in $[1, d]$ increments gradually, with each increment representing a new search for $R_{best,i}$ and an evaluation of its fitness value. If $f_R < f_b$, indicating that a better result has been obtained through local search, $R_{best,i}$ will replace the original $X^t_{best,i}$, and f_R will replace f_b to guide the next generation with the new global optimal fitness value. If $f_R \geq f_b$, indicating that no better result has been found, $X^t_{best,i}$ retains its original value.

3.4. Floyd path optimization

The ESSA generates optimal paths based on the grid map, but these paths often contain unnecessary turning points. To address this, We propose a path smoothing optimization method based on the Floyd algorithm to reduce redundant nodes along the path. The process is as follows:

Step 1: From the initial point, identify segments where three or more consecutive nodes are collinear. If the slope of the line connecting these nodes is either zero or undefined, remove the intermediate nodes, retaining only the first and last nodes of each collinear segment. Add the retained nodes to a new list.

Step 2: Starting from the first node, consider the line connecting it to the next node as a candidate path. If the distance from this path to the nearest obstacle exceeds the robot's expansion radius, remove the intermediate node and replace the path with a direct connection. If the path is obstructed, retain the intermediate node and continue to the next. This process is repeated until the last node is reached.

This optimization method eliminates unnecessary turning points, ensuring smoother, more efficient paths while avoiding obstacles. Step 2 is mathematically described as

$$list = list + \{n\,[k-1]\}\,, d_{ik} < R, k > 1, k > i \tag{10}$$

where *list* is the new list, $n[k-1]$ represents the $(k-1)$-th node, d_{ik} is the straight-line distance between the current node i and the k-th node, and R is the robot's expansion radius.

As shown in Figure 2, after Step 1, nodes (2, 4, 6, 8, 10, 11) are removed. After Step 2, nodes (3, 5, 7, 13) are removed, and nodes (1, 9, 12, 14) are selected as the new path nodes.

As shown in Figure 3, the red dots represent the search nodes. The left panel displays the original path, while the right panel shows the optimized path. According to Table I, the optimized path is, on average, 0.9% shorter in length, and the number of search nodes is reduced by 43.75% compared to the original path.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

Table I. *Performance comparison of the path before and after optimization.*

Algorithm	Distance (m)	Number of nodes
Before path optimization	44.746	16
After path optimization	44.348	9

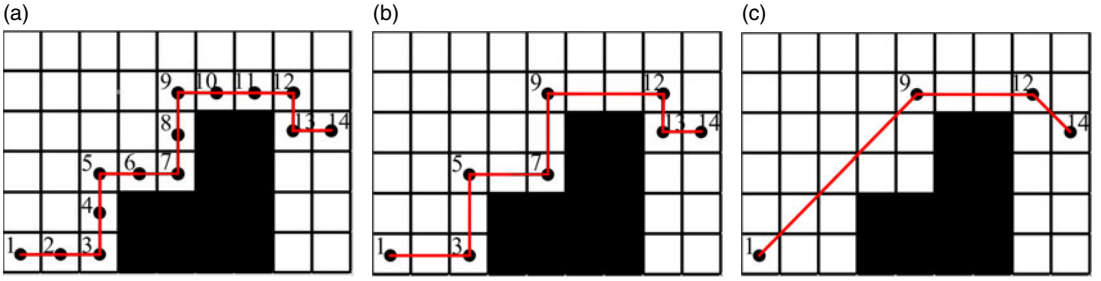


Figure 2. *Principle of the floyd path optimization.*

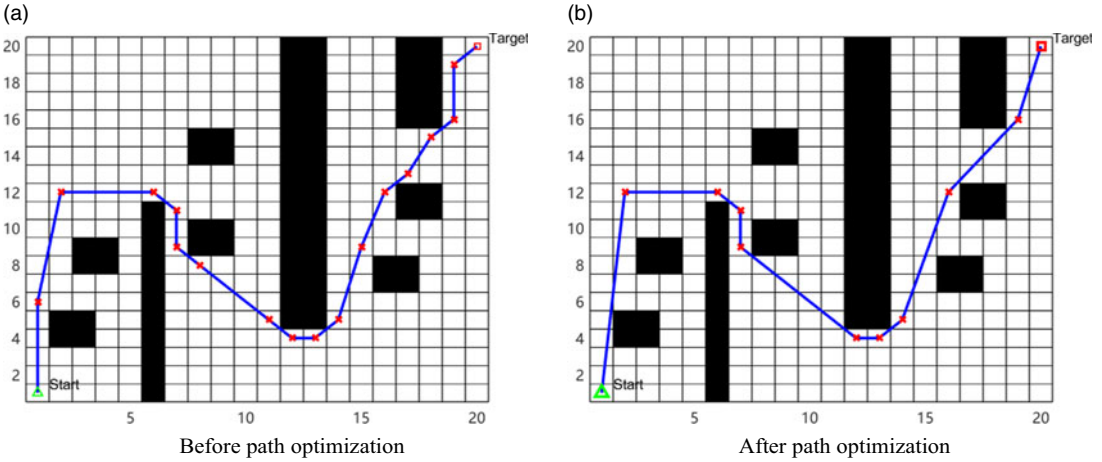


Figure 3. *Experiment on removing redundant nodes using the Floyd algorithm.*

4. Local path planning algorithm based on DWA and fusion algorithm

4.1. DWA principle

Assume that the mobile robot moves in a straight line with omnidirectional motion over a very short time, with a linear velocity v and an angular velocity ω. The robot's motion trajectory can be expressed by the following Eq. (11):

$$\begin{cases} x = x + v_x \Delta t \cos\theta_t - v_y \Delta t \sin\theta_t \\ y = y + v_x \Delta t \sin\theta_t + v_y \Delta t \cos\theta_t \\ \theta_t = \theta_t + \omega \Delta t \end{cases} \tag{11}$$

where v and ω represent the linear velocity and angular velocity, and Δt is the simulation prediction time interval.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

The selection of the actual velocity of the mobile robot is constrained by its inherent physical properties and real-time external environmental factors, facing three main constraints:

(1) Velocity range limits
Considering the hardware limitations of the robot and the extreme values of speed, the highest and lowest velocity limits that the robot can achieve are defined. Thus, the velocity set can be expressed as

$$V_m = \{(v, \omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{12}$$

where $v_{\min}$ and $v_{\max}$ are the minimum and maximum linear velocities, and $\omega_{\min}$ and $\omega_{\max}$ are the minimum and maximum angular velocities.

(2) Acceleration and deceleration constraints
Due to the torque limitations of the motors and their driving components, the acceleration and deceleration of the robot are limited. Within the given prediction time, the extreme values of the velocity combinations can be determined by the maximum acceleration and deceleration. Thus, the velocity set can be expressed as

$$V_d = \{(v, \omega) | v \in [v_t - \dot{v}\Delta t, v_t + \dot{v}\Delta t], \omega \in [\omega_t - \dot{\omega}\Delta t, \omega_t + \dot{\omega}\Delta t]\} \tag{13}$$

Where v_t and ω_t are the linear and angular velocities at time t, $\dot{v}$, and $\dot{\omega}$ are the maximum linear and angular accelerations of the mobile robot, and Δt is the sampling interval.

(3) Braking safety distance
During path planning, it is essential to ensure that the mobile robot remains collision-free from obstacles, especially under maximum deceleration. Sufficient braking distance must be ensured to avoid collisions before the robot's speed reaches zero. Therefore, considering the braking safety distance, the velocity set can be defined as the set of velocities that meet this constraint, expressed as

$$V_a = \left\{(v, \omega) | v \leq (2dist(v, \omega)\dot{v})^{\frac{1}{2}}, \omega \leq (2dist(v, \omega)\dot{\omega})^{\frac{1}{2}}\right\} \tag{14}$$

Where $dist(v, \omega)$ is the evaluation function representing the shortest distance from the mobile robot to the obstacle.

4.2. Evaluation function

The DWA algorithm evaluates the robot's ability to avoid obstacles while efficiently approaching the target. The optimal speed vector, corresponding to the best motion trajectory, is selected based on this evaluation. Once determined, the robot moves toward the target at this speed. The evaluation function is defined as

$$G(v, \omega) = \alpha \cdot head(v, \omega) + \beta \cdot vel(v, \omega) + \gamma \cdot dist(v, \omega) \tag{15}$$

where $head(v, \omega)$ represents the orientation angle towards the target, $vel(v, \omega)$ represents the current speed value, and $dist(v, \omega)$ represents the shortest safe distance to the nearest obstacle.

In the final evaluation stage, the deviation angle between the robot's final orientation and the target direction is considered. Minimizing this angle increases the trajectory's proximity to the target, so a smaller deviation angle receives a higher weight:

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

$$head(v,\omega) = \frac{\pi - \Delta\theta_i}{\sum_{i=1}^{N}(\pi - \Delta\theta_i)} \tag{16}$$

where N is the total number of simulated trajectories, i is the i-th simulated trajectory, and $\Delta\theta_i$ is the deviation angle between the end posture of the i-th simulated trajectory and the target orientation.

The robot's instantaneous speed is a critical factor in performance evaluation. As speed increases, its weight in the evaluation function rises, improving the overall score:

$$vel(v,\omega) = \frac{v_i}{\sum_{i=1}^{N} v_i} \tag{17}$$

where v_i represents the absolute value of the speed at the end of the i-th simulated trajectory.

For obstacle avoidance, the minimum distance to the nearest obstacle $dist(v,\omega)$ plays a key role. As this distance decreases, its weight in the evaluation function decreases, lowering the overall score:

$$dist(v,\omega) = \frac{d_i}{\sum_{i=1}^{N} d_i}\left(d_i = \begin{cases} d_i, d_i < D_0 \\ D_0, d_i \geq D_0 \end{cases}\right) \tag{18}$$

where d_i represents the closest distance to the obstacle at the end of the i-th simulated trajectory, and D_0 is a constant.

4.3. ESSA-DWA fusion algorithm

4.3.1. Algorithm fusion

The DWA has good dynamic obstacle avoidance performance but still suffers from issues such as the lack of global prior information, resulting in nonoptimal paths or even an inability to reach the target point. In this article, the ESSA is proposed, which can acquire global optimal path information in real time but is only applicable for obstacle avoidance in static environments. To address the limitations of the aforementioned algorithms, this study combines the advantages of both approaches and proposes a novel dynamic path planning method for mobile robots, called ESSA-DWA.

The main steps of the algorithm are as follows: First, the ESSA-related parameters are set, the population is initialized using the Tent chaotic mapping, dynamic inertial weight is updated to adjust the position of the producers, and a local search is performed on the optimal solution to identify the individual with the best fitness value, thereby generating a globally optimal path. Second, the Floyd algorithm is applied to optimize the current path by removing redundant nodes, generating a smooth optimal path, and extracting the path nodes as key nodes for the DWA algorithm. Finally, the DWA-related parameters are set, and the key nodes are incorporated into the DWA algorithm as references for local planning, ultimately generating a smooth trajectory that realizes both global optimal path planning and local dynamic obstacle avoidance. The flowchart of the integrated algorithm is shown in Figure 4.

4.3.2. Complexity analysis

The time complexity of the ESSA algorithm can be represented as $O(G \times N \times D)$, where G is the maximum number of iterations, N is the population size, and D is the path dimension. The runtime of the algorithm is proportional to the product of these three factors. In each iteration, the following operations are performed: Tent chaotic initialization, dynamic inertia weight adjustment, and local search.

The time complexity of the Floyd path smoothing algorithm can be represented as $O(k^2)$, where k is the number of path nodes. The algorithm consists of two-stage optimization: the first stage traverses the path nodes to remove redundant nodes, and the second stage removes nodes by evaluating their distance to obstacles. For example, in a 20×20 map, the value of k typically ranges between 10 and 20, and the algorithm's runtime grows quadratically with respect to k.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

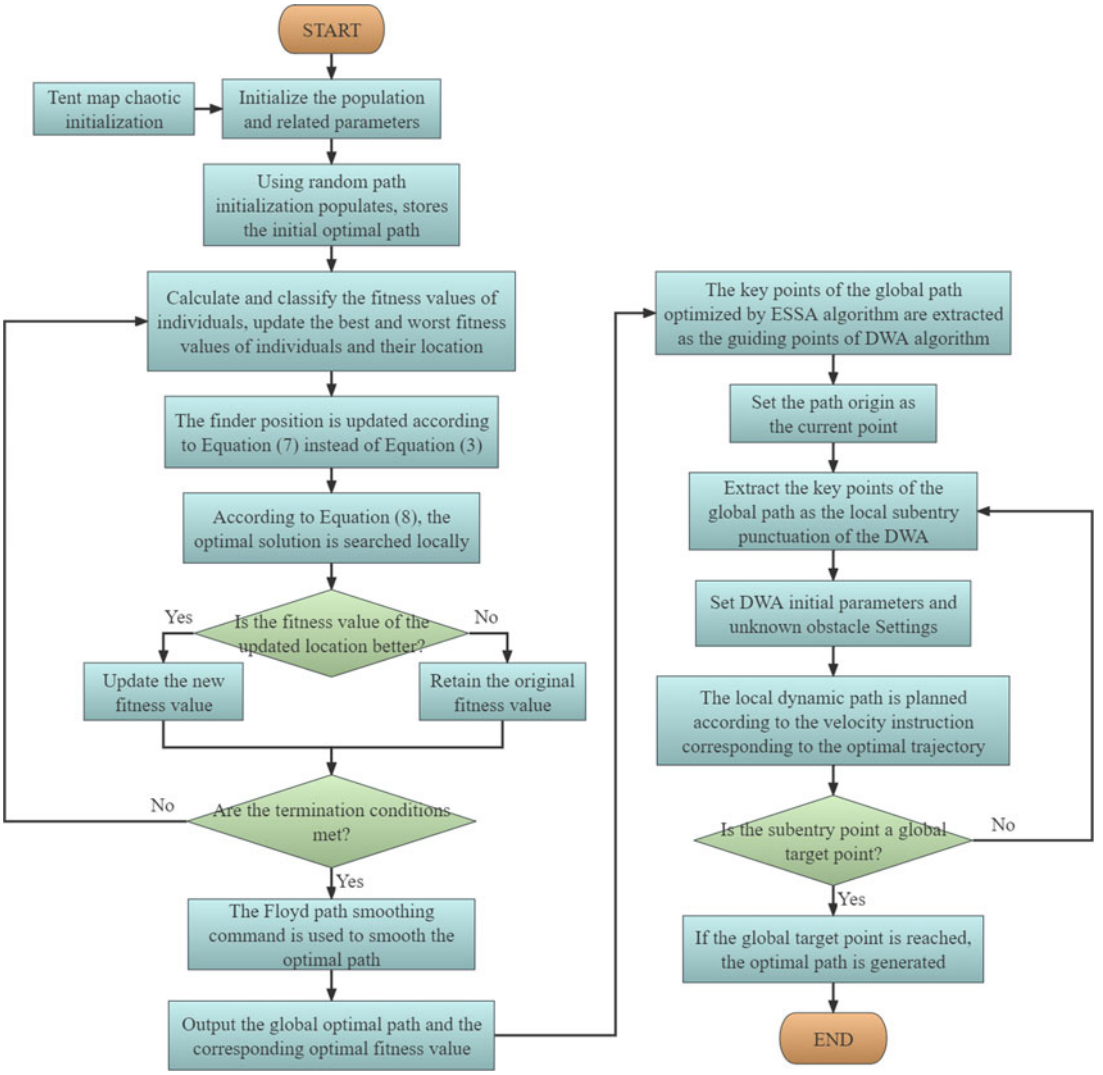


Figure 4. *The flowchart of the ESSA-DWA fusion algorithm.*

The time complexity of the DWA algorithm is $O(n^2)$, where n represents the number of optimization variables. As the number of optimization variables increases, the runtime of the algorithm grows quadratically. In this article, $n = 4$, corresponding to the current robot position, heading angle, linear velocity, and angular velocity.

Based on the above analyses, the time complexity of the ESSA-DWA fusion algorithm is $O(G \times N \times D + k^2 + n^2)$.

Compared to traditional SSA, the ESSA algorithm has a higher time complexity. For complex maps with numerous path nodes, the runtime may increase. However, through improvements such as Tent chaotic initialization, dynamic inertia weight adjustment, and local search, combined with the DWA algorithm, the ESSA algorithm provides shorter path lengths, smoother paths, faster convergence, better adaptability to different environments, and enhanced obstacle avoidance performance.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

The pseudocode for the ESSA-DWA integrated algorithm is as follows:

Algorithm: ESSA-DWA Fusion Algorithm

Input:
ESSA: Maximum number of iterations (*max_ gen*); Population size (n); Safety threshold (*soft_value*); Number of producers (*recover_Num*); Ratio of scouts (*scout_Num*).
DWA: Robot speed parameters; Evaluation function coefficients (α, β, γ).
Output: Optimal path (*NewPath*2); Driving trajectory (*Result_x*).

Initialize the sparrow population using Tent chaotic mapping;
While ($t < max_ gen$)
Calculate the fitness values of the population and select the sparrow with the lowest fitness as the current best solution;
for $i = 1:$ *recover_Num*
Update the position of the producers using Eq. (7).
end for
for $i = (recover_Num + 1): n$
Update the position of the followers using Eq. (4);
end for
for $i = 1:$ *scout_Num*
Update the position of the scouts using Eq. (5);
end for
for $i = 1: n$
Perform local search on the best solution using Eq. (8) to find the individual with the optimal fitness value;
end for
$t = t + 1$;
end While
Obtain the global optimal path (*Path*) from **ESSA**;
Smooth the path (*Path*) using the Floyd path smoothing method;
While (true)
Remove intermediate redundant nodes that are collinear to obtain *NewPath*;
Update the path using Eq. (10), checking the distance between the path nodes and obstacles to obtain *NewPath*2;
end While
Set the robot speed parameters and evaluation function coefficients (α, β, γ) for **DWA**;
Extract key nodes from the optimized global optimal path (*NewPath*2);
Integrate the key point information into the local planner *DWA_ct_dong*;
Control the robot to follow the global optimal path;
Return the smoothed optimal driving trajectory (*Result_x*).

5. Simulation experiments and practical experiments

5.1. Simulation experimental environment

To ensure the completeness and comparability of the experimental results, all simulation experiments follow a uniform configuration. A grid-based method was used to construct the grid map for the simulation experiments. The experimental environment for this study consists of a desktop computer with the following specifications: AMD Ryzen 5 5600 6-Core Processor @ 3.50 GHz, 16 GB of RAM, and a 64-bit Windows 10 operating system. The simulation software used is MATLAB R2021a.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

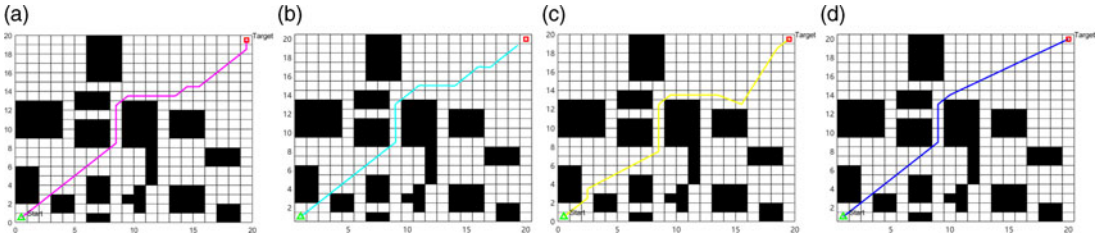


Figure 5. *Planned paths of the four algorithms in environment model 1.*

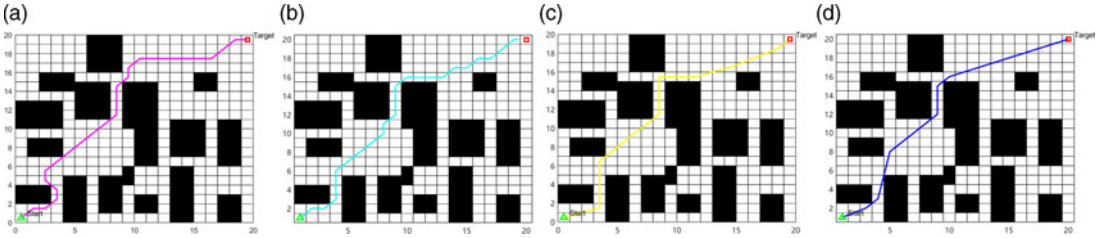


Figure 6. *Planned paths of the four algorithms in environment model 2.*

5.2. Simulation experiments and analysis of global path planning using the ESSA algorithm

To assess the efficiency and generality of the ESSA algorithm for static path planning, experiments were conducted on grid maps of sizes 20×20, 30×30, and 50×50, each configured with three different environment models of varying complexity. The performance of the ESSA algorithm was compared in detail with the ACO, GWO, and SSA algorithms across various scenarios. ACO and GWO were chosen as benchmarks due to their extensive use and research in path planning. Each algorithm was subjected to 30 simulation experiments in each environmental model. The performance evaluation primarily focused on the average path length and mean execution time, along with their standard deviations (Sd.).

In the simulations, each grid cell had a side length of 1 m. The starting point was marked with a green triangle and the target point with a red square. Obstacles were represented by black grid cells and walkable areas by white grid cells. The parameters for the algorithms were as follows:

ACO: 200 iterations, ant population size of 50, pheromone factor of 2, heuristic factor of 6, pheromone evaporation coefficient of 0.1, and pheromone increment coefficient of 5.

GWO: 200 iterations, grey wolf population size of 50.

SSA: 200 iterations, sparrow population size of 50, safety value of 0.8, 30% producers, and 20% scouts.

In Figures 5–13, the paths planned by the different algorithms are shown:

(1) ACO algorithm (purple solid line);
(2) GWO algorithm (cyan solid line);
(3) SSA algorithm (yellow solid line);
(4) ESSA algorithm (blue solid line).

The initial point is represented by a green triangle and the target point by a red square.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

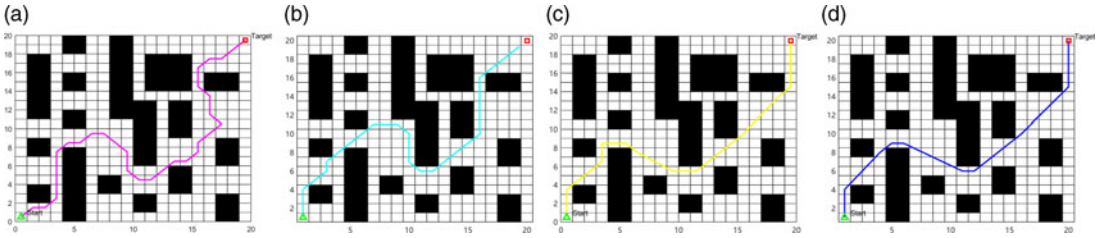


Figure 7. *Planned paths of the four algorithms in environment model 3.*

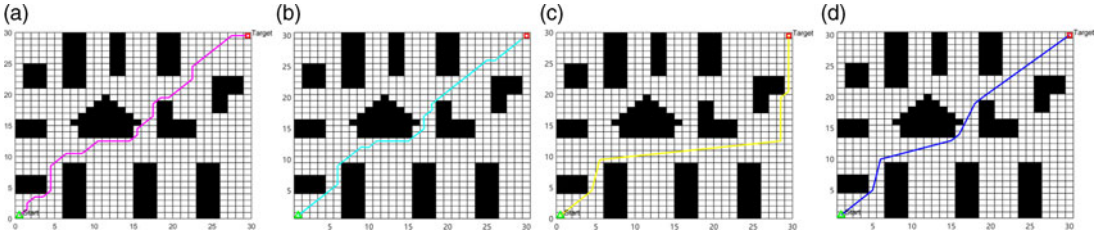


Figure 8. *Planned paths of the four algorithms in environment model 4.*

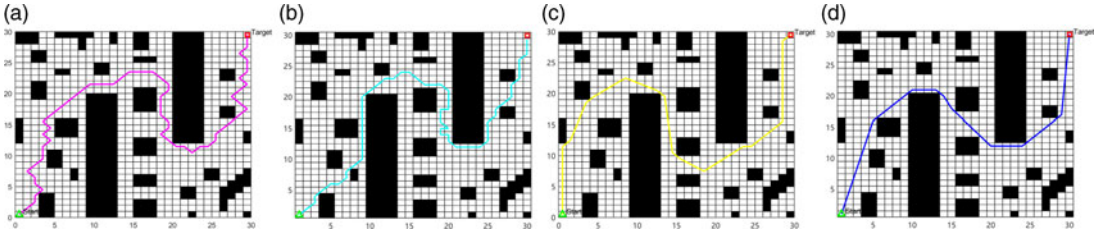


Figure 9. *Planned paths of the four algorithms in environment model 5.*

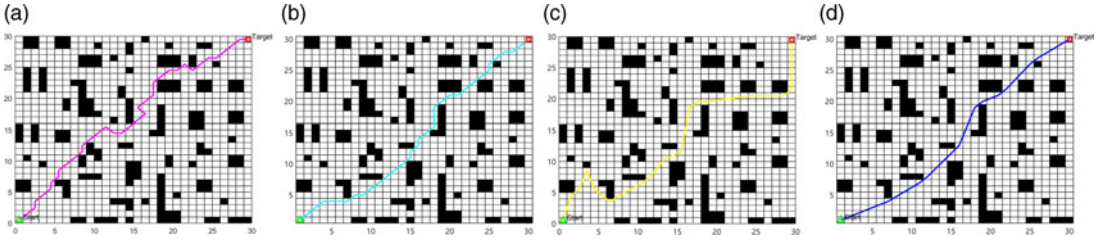


Figure 10. *Planned paths of the four algorithms in environment model 6.*

5.2.1. 20 × 20 grid map

To more accurately represent the real-world environment of mobile robots, this study replicated the environmental models used in ref. [37] on a 20 × 20 grid map and compared the proposed ESSA algorithm with the improved SSA algorithm from ref. [37] to highlight the superior performance of the proposed approach. Figures 5–7 illustrate the global path planning results of ACO, GWO, SSA, and ESSA for environment models 1–3, respectively. These three environmental models vary in type and progressively increase in complexity.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

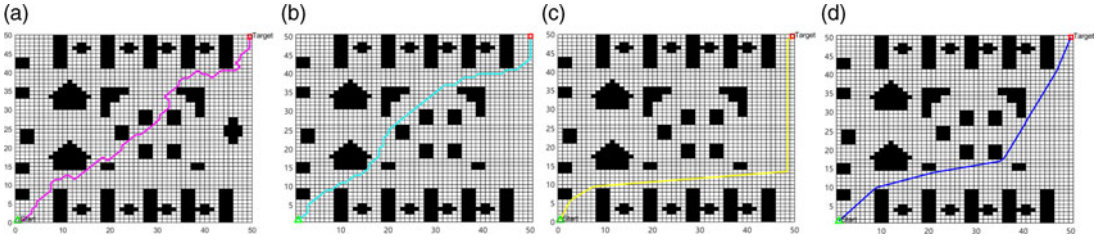


Figure 11. *Planned paths of the four algorithms in environment model 7.*

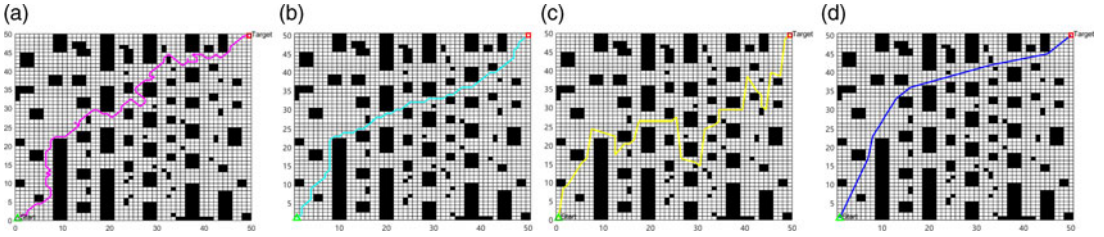


Figure 12. *Planned paths of the four algorithms in environment model 8.*

Figures 5–7 present the global path planning results of ACO, GWO, SSA, and ESSA across Environment Models 1, 2, and 3. By examining and comparing the results for the three environment models, it is clearly evident that as the complexity of obstacle distribution increases, the superior performance of the ESSA algorithm becomes more pronounced. The ESSA algorithm is capable of efficiently planning the optimal path, and, compared to other algorithms, the resulting path is smoother.

Table II shows that the ESSA algorithm outperforms the other algorithms across all three environmental models on the 20×20 grid map. When compared to ACO, GWO, SSA, and ISSA [37], in Environment Model 1, the ESSA algorithm reduced the average path length by 12.44%, 2.82%, 13.22%, and 0.17%, respectively, while the average execution time was reduced by 66.25%, 86.40%, 37.74%, and 27.63%. In Environment Model 2, the ESSA algorithm reduced the average path length by 11.34%, 4.45%, 4.51%, and 1.34%, respectively, while the average execution time was reduced by 74.06%, 90.83%, 52.03%, and 47.88%. In Environment Model 3, the ESSA algorithm reduced the average path length by 13.80%, 11.19%, 9.08%, and 0.68%, respectively, while the average execution time was reduced by 70.98%, 89.59%, 45.65%, and 28.64%.

The experimental results demonstrate that the ESSA algorithm proposed in this article exhibits excellent convergence performance on the 20×20 grid map, generating the optimal and smoother paths. This also confirms that the ESSA algorithm effectively enhances the traditional SSA algorithm and outperforms the other four heuristic algorithms.

5.2.2. 30 × 30 grid map

To demonstrate the superiority of the proposed algorithm in various complex environments, three additional environmental models were created on a 30×30 grid map, and comparisons were made among the ACO, GWO, SSA, and ESSA algorithms. Figures 8–10 illustrate the global path planning results for ACO, GWO, SSA, and ESSA in environment models 4–6, respectively. These environmental models differ in type and progressively increase in complexity.

Figures 8–10 illustrate the global path planning results of the ACO, GWO, SSA, and ESSA algorithms in Environment Models 4, 5, and 6. In these three environment models, the paths planned by the ESSA algorithm are notably superior to those of the other algorithms, exhibiting smoother trajectories and fewer turning points.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

Table II. *Performance comparison of various algorithms in environment models 1–3 on a 20 × 20 grid map (30 simulations).*

20 × 20 grid map	Algorithm	Path length (m)		Execution time (s)	
		Average	Sd.	Mean	Sd.
Environment model 1	ACO	32.425	2.098	2.324	0.201
	GWO	29.213	0	5.723	0.528
	SSA	32.716	1.267	1.291	0.115
	ISSA [37]	28.438	/	1.075	/
	ESSA	**28.390**	**0**	**0.778**	**0.022**
Environment model 2	ACO	33.376	1.864	2.325	0.142
	GWO	30.971	0	6.578	0.049
	SSA	30.988	0.680	1.257	0.092
	ISSA [37]	29.992	/	1.157	/
	ESSA	**29.591**	**0**	**0.603**	**0.012**
Environment model 3	ACO	39.799	2.367	2.757	0.242
	GWO	38.627	0	7.682	0.060
	SSA	37.932	1.472	1.227	0.106
	ISSA [37]	34.541	/	1.121	/
	ESSA	**34.305**	**0**	**0.800**	**0.013**

Table III. *Performance comparison of various algorithms in environment models 4–6 on a 30 × 30 grid map (30 simulations).*

30 × 30 grid map	Algorithm	Path length (m)		Execution time (s)	
		Average	Sd.	Mean	Sd.
Environment model 4	ACO	52.350	4.004	4.567	0.552
	GWO	44.527	0	12.351	0.113
	SSA	49.356	2.054	2.493	0.290
	ESSA	**43.322**	**0**	**2.168**	**0.017**
Environment model 5	ACO	84.443	7.301	4.696	0.401
	GWO	73.113	0	20.441	0.199
	SSA	76.425	5.166	2.575	0.236
	ESSA	**61.165**	**0**	**2.016**	**0.033**
Environment model 6	ACO	51.583	2.760	5.165	0.380
	GWO	44.527	0	13.589	0.117
	SSA	68.852	5.223	2.819	0.184
	ESSA	**42.329**	**0**	**2.159**	**0.025**

Table III demonstrates that the ESSA algorithm outperforms the other algorithms across all three environment models on a 30 × 30 grid map. As the map size increases, the average planning time for all algorithms increases. Specifically, in Environment Model 4, the average path length generated by ESSA is reduced by 17.25%, 2.71%, and 12.23% compared to ACO, GWO, and SSA, respectively, while the average execution time is reduced by 52.53%, 82.45%, and 13.04%. In Environment Model 5, ESSA reduces the average path length by 27.57%, 16.34%, and 19.94%, and the average execution time by 57.09%, 90.14%, and 21.71%, respectively. In Environment Model 6, the average path length is reduced by 17.94%, 4.94%, and 38.51%, and the average execution time is reduced by 58.20%, 84.11%, and 23.41%, respectively.

The experimental results indicate that, as environmental complexity and path length increase, the proposed ESSA algorithm maintains a significant advantage on 30 × 30 grid maps.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

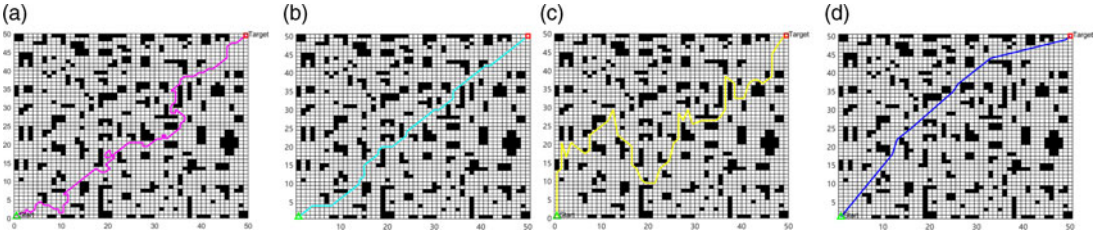


Figure 13. *Planned paths of the four algorithms in environment model 9.*

Table IV. *Performance comparison of various algorithms in environment models 7–9 on a 50 × 50 grid map (30 simulations).*

500 × 50 grid map	**Algorithm**	**Path length (m)**		**Execution time (s)**	
		Average	**Sd.**	**Mean**	**Sd.**
Environment model 7	ACO	111.470	8.241	10.026	0.901
	GWO	77.498	0	37.509	0.694
	SSA	89.788	1.346	**4.119**	0.217
	ESSA	**75.407**	**0**	6.952	**0.075**
Environment model 8	ACO	114.495	10.518	8.210	0.713
	GWO	79.255	0	37.809	0.826
	SSA	157.668	12.293	**7.327**	0.554
	ESSA	**76.062**	**0**	8.198	**0.061**
Environment model 9	ACO	103.088	9.885	10.609	0.873
	GWO	73.983	0	33.312	0.432
	SSA	168.597	12.635	7.831	0.586
	ESSA	**72.381**	**0**	**6.177**	**0.073**

5.2.3. 50 × 50 grid map

To demonstrate the general applicability of the proposed algorithm, this study also conducted simulation experiments on a 50 × 50 grid map. Three different environment models were used for comparison on the 50 × 50 grid map. Figures 11–13 illustrate the global path planning results of ACO, GWO, SSA and ESSA for environment models 7–9, respectively. Each of the three environment models varies in type and progressively increases in complexity.

Figures 11–13 present the global path planning results of the ACO, GWO, SSA, and ESSA algorithms in Environment Models 7, 8, and 9. In these three environment models, ESSA demonstrates strong adaptability to complex environments, with superior global search capabilities. The trajectories generated by ESSA exhibit fewer turning points and are closer to straight lines, resulting in shorter and more efficient paths.

Table IV shows that the ESSA algorithm outperforms the other algorithms across all three environment models on a 50 × 50 grid map. Specifically, in Environment Model 7, the average path length generated by ESSA is reduced by 32.35%, 2.71%, and 16.03% compared to ACO, GWO, and SSA, respectively. However, the average execution time is reduced by 31.89% and 81.47% for ACO and GWO but increased by 68.78% for SSA. In Environment Model 8, ESSA reduces the average path length by 33.62%, 4.10%, and 51.79%, with the average execution time reduced by 0.15% and 78.31% for ACO and GWO but increased by 11.88% for SSA. In Environment Model 9, the average path length is reduced by 29.78%, 2.17%, and 57.07%, while the average execution time is reduced by 41.77%, 81.46%, and 21.12%, respectively.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

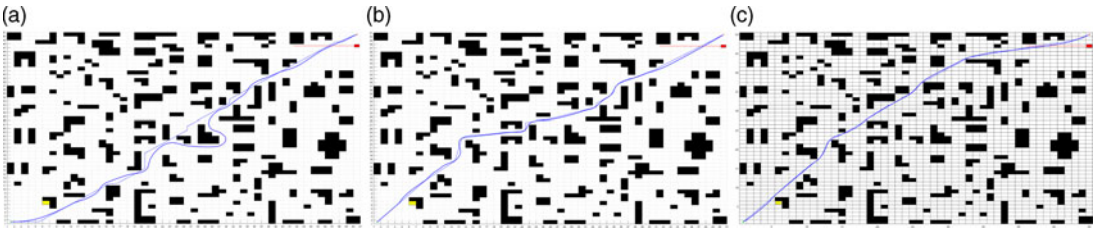


Figure 14. *Comparison of fusion algorithms.*

The experimental results indicate that, as the map size increases, the proposed ESSA algorithm continues to perform well in complex environments on 50×50 grid maps, generating shorter average path lengths. Due to the increased number of path nodes, the time complexity of path smoothing also rises, leading to longer execution times. Nevertheless, this does not affect ESSA's excellent performance in other aspects within complex environments.

In summary, ESSA successfully identifies optimal paths on 20×20, 30×30, and 50×50 grid maps. Compared to ACO, GWO, and SSA, ESSA exhibits significant advantages in terms of path length and execution time. In complex environments, ESSA demonstrates stronger adaptability and enhanced global search capabilities, generating smoother optimal paths. However, as the number of path nodes increases, the time complexity also rises, resulting in longer execution times—an aspect that requires further improvement. Overall, the results demonstrate that ESSA performs exceptionally well in finding globally optimal paths in complex environments across different map sizes.

5.3. Simulation experiments and analysis of local dynamic obstacle avoidance using the ESSA-DWA fusion algorithm

5.3.1. Dynamic obstacle avoidance experiment and analysis of the fusion algorithm

Following the demonstration of ESSA's superiority in global path planning, this study further integrates the ESSA algorithm with the DWA to highlight the dynamic obstacle avoidance capabilities of the ESSA-DWA fusion algorithm. For the DWA algorithm, the following parameter settings were used: the maximum linear velocity was set to 0.5 m/s, the maximum linear acceleration to 1 m/s^2, the maximum angular velocity to 30°/s, and the maximum angular acceleration to 50°/s^2. The velocity resolution was set to 0.1 m/s, the angular velocity resolution to 3.0°/s, and the prediction horizon to 3.0 s. The parameters of the evaluation function were defined as $\alpha = 0.05$, $\beta = 0.2$, and $\gamma = 0.1$. The robot's inflation radius was set to 0.5 m, and the movement speed of dynamic obstacles was set to 0.1 m/s. In the simulation environment, yellow grid cells represent unknown static objects, while red grid cells indicate dynamic obstacles, with red arrows denoting the direction of movement for dynamic obstacles. The starting grid is marked with a triangle, and the goal grid is marked with a circle. The overall planned path is represented by a dashed line, while the locally generated obstacle-avoidance path is indicated by a solid line.

Additionally, this study conducts a comparative analysis of the ESSA-DWA fusion algorithm against the traditional A*-DWA and an improved A*-DWA from ref. [13] under identical environmental conditions and obstacle configurations, aiming to highlight the superior performance of the ESSA-DWA approach. The performance comparison of the evaluated algorithms is visually depicted in Figure 14.

As illustrated in Figure 14, the planned paths generated by traditional A*-DWA and improved A*-DWA, and ESSA-DWA differ significantly under identical environmental conditions. According to Table V, under the same obstacle configurations, the ESSA-DWA algorithm achieves notable improvements in both travel distance and execution time. Specifically, compared to A*-DWA and improved A*-DWA, ESSA-DWA reduces the travel distance by 8.05% and 2.27%, respectively, while shortening the execution time by 8.60% and 4.62%, respectively. The experimental results demonstrate that the

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

Table V. *Performance comparison of the three fusion algorithms.*

Algorithm	Distance (m)	Time (s)
A*-DWA	78.71	162.7
Improve A*-DWA [13]	74.05	155.9
ESSA-DWA	72.37	148.7

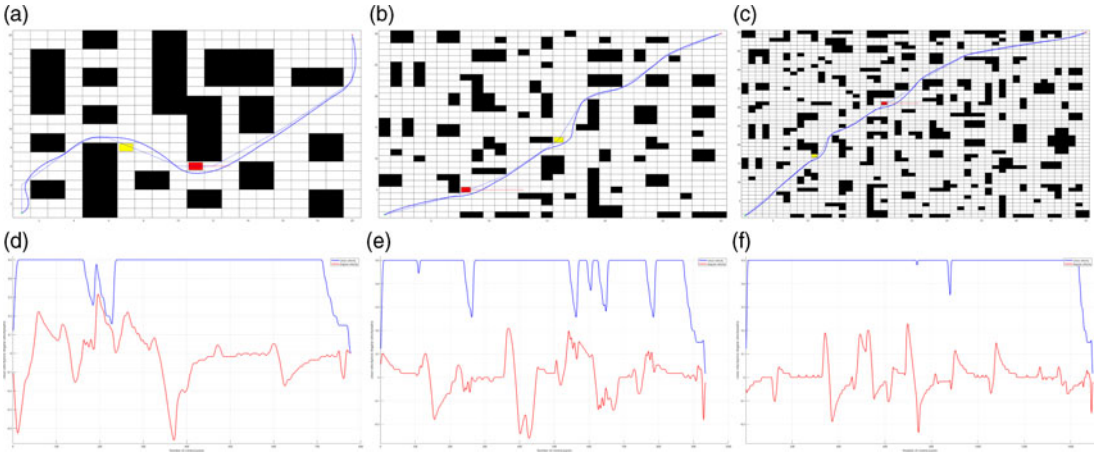


Figure 15. *Dynamic obstacle avoidance simulation and corresponding linear and angular velocities.*

proposed ESSA-DWA fusion algorithm exhibits superior dynamic adaptability and overall performance, making it a highly effective solution for navigation in complex and dynamic environments.

5.3.2. Dynamic obstacle avoidance experiment and analysis of the fusion algorithm in different maps

To evaluate the adaptability of the ESSA-DWA fusion algorithm across different environments, this study selects one environment from each of the 20×20, 30×30, and 50×50 grid maps in the simulation experiments, specifically Environment Models 3, 6, and 9. The performance of the ESSA-DWA fusion algorithm was assessed under these varying map conditions. In scenarios involving randomly introduced obstacles, the algorithm was tested to analyze its ability to handle unpredictable environmental changes. The results of the dynamic obstacle avoidance simulations, including the corresponding linear and angular velocity profiles, are illustrated in Figure 15. These results demonstrate the algorithm's capability to maintain smooth and efficient navigation while dynamically adapting to the presence of random obstacles and varying map complexities.

Figure 15 illustrates the dynamic obstacle avoidance capabilities of the ESSA-DWA fusion algorithm in Environment Models 3, 6, and 9, along with the corresponding linear and angular velocity profiles. It can be observed that, under different environmental models and in the presence of unknown obstacles, the mobile robot successfully avoids these obstacles and reaches the target destination. Table VI summarizes the performance metrics of the three sets of dynamic obstacle avoidance simulations. The experimental results demonstrate that the proposed ESSA-DWA fusion algorithm effectively achieves real-time obstacle avoidance across varying environments.

In summary, the results of the three sets of dynamic obstacle avoidance experiments conducted under different environmental conditions validate the performance of the hybrid algorithm across diverse scenarios. These findings highlight the feasibility, adaptability, and practicality of the ESSA-DWA fusion algorithm.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

Table VI. *Dynamic obstacle avoidance performance of ESSA-DWA in three environmental models.*

Map	Distance (m)	Time (s)
Environment Model 3	35.54	77.6
Environment Model 6	43.02	93.3
Environment Model 9	72.79	149.4

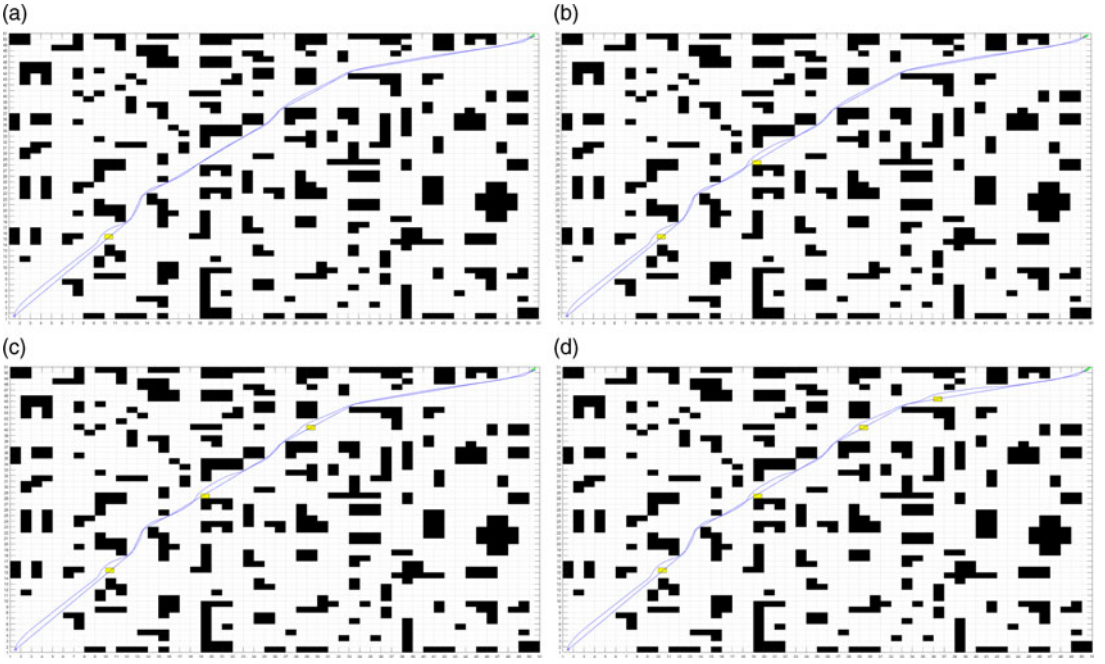


Figure 16. *Dynamic obstacle avoidance simulation with newly added unknown obstacles.*

5.3.3. Randomly added obstacle experiment and analysis

In the absence of newly introduced unknown obstacles, the ESSA-DWA fusion algorithm is capable of successfully achieving the optimal path. To thoroughly evaluate the obstacle avoidance performance of the ESSA-DWA fusion algorithm in dynamic environments, simulations were conducted on a 50×50 grid map using four environmental models, with Environment Model 9 serving as a representative example. A number of random unknown obstacles were added to the experimental environment. The starting and ending grids were denoted by a triangle and a circle, respectively, while green cursors indicated the robot's directional sensing. Unpredictable obstacles were represented by gray rectangles. The path generated by the ESSA algorithm was shown with dashed lines, while the real-time navigation of the fusion algorithm was depicted with solid lines, highlighting the algorithm's flexibility in rapidly adapting to dynamic changes in the environment.

As shown in Figure 16, the robot effectively navigates around unknown obstacles while following the path generated by the ESSA algorithm. As the number of randomly placed obstacles increases, the planned route consistently demonstrates its effectiveness in both circumventing obstacles and ensuring successful navigation to the predetermined destination. The hybrid algorithm efficiently handles random obstacles within grid maps, maintaining a smooth trajectory that remains closely aligned with the globally optimal path. When multiple unknown obstacles are introduced in Environment Model 9, the performance summary of the ESSA-DWA fusion algorithm is presented in Table VII.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

Table VII. *Dynamic obstacle avoidance performance of the ESSA-DWA fusion algorithm under newly added unknown obstacles in environment Model 9.*

Number of random obstacles	Distance (m)	Time (s)
One obstacle	72.48	148.1
Two obstacles	72.73	148.7
Three obstacles	72.78	148.7
Four obstacles	73.02	149.2

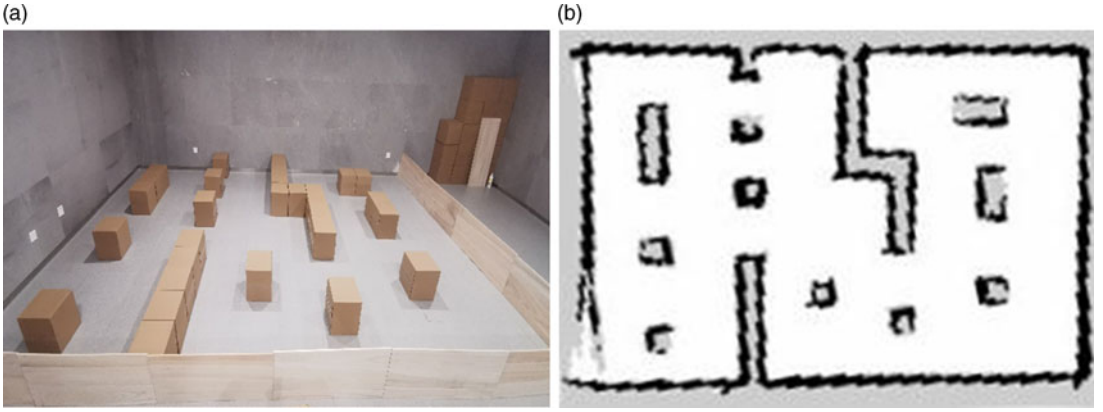


Figure 17. *Experimental environment.*

Based on the simulation results across four different environments, the path generated by the proposed ESSA-DWA hybrid algorithm is largely consistent with the path planned in static environments. In scenarios involving unknown obstacles, the algorithm successfully circumvents these obstacles, producing a smooth trajectory that effectively balances global optimality and obstacle avoidance. This highlights the algorithm's capability to perform real-time local dynamic path planning based on the global path. Furthermore, the ESSA-DWA hybrid algorithm seamlessly adapts to complex dynamic obstacle environments, thereby enhancing the operational efficiency of the robot.

5.4. Simulation experiments and analysis of the ESSA-DWA fusion algorithm

To evaluate the proposed algorithm in real-world applications, practical path planning experiments were conducted using an Mecanum wheel mobile robot (XTARK) on an Ubuntu 18.04 system. The Cartographer-SLAM algorithm generated the cost map for the environment, and the ROS visualization tool, Rviz, was used to display it. In the cost map, white areas represent walkable regions, black areas indicate obstacles, and gray areas show obstacle boundaries. For the robot parameters, the settings for both ESSA and DWA were consistent with those used in the simulation experiments, with the robot's inflation radius set to 0.2 m. Due to experimental constraints, a simplified environment based on Environment Model 3 was created using cardboard boxes. The experimental setup is shown in Figure 17.

The real-world experiments were conducted in two phases: static and dynamic path planning. In the static path planning experiment, the initial and target positions of the robot were predefined, and the ESSA-DWA fusion algorithm was utilized to compute the robot's path from the starting point to the target within a static environment. Throughout the navigation process, the robot adhered to the planned trajectory. As depicted in Figure 18, the robot successfully departed from its initial position and skillfully maneuvered around all static obstacles encountered along its path. Ultimately, the robot safely reached the target point by following the globally optimal route.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

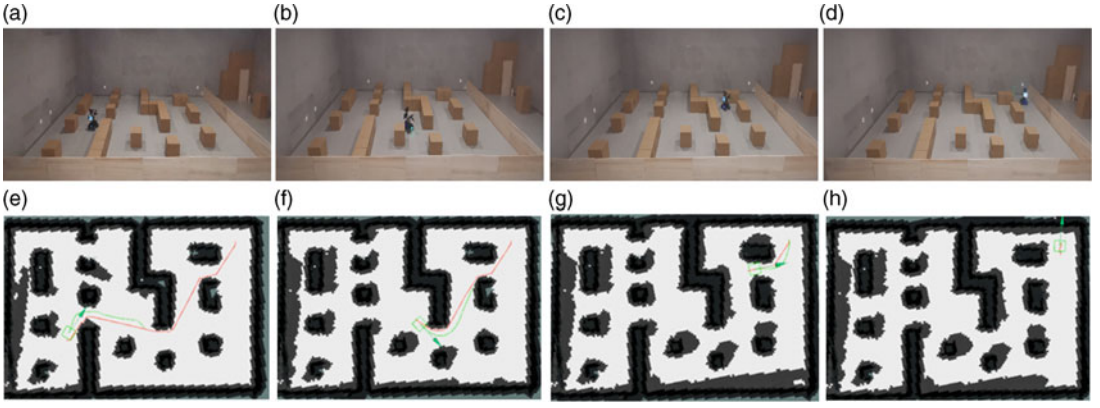


Figure 18. *Obstacle avoidance process in static environment and their Rviz in ROS.*

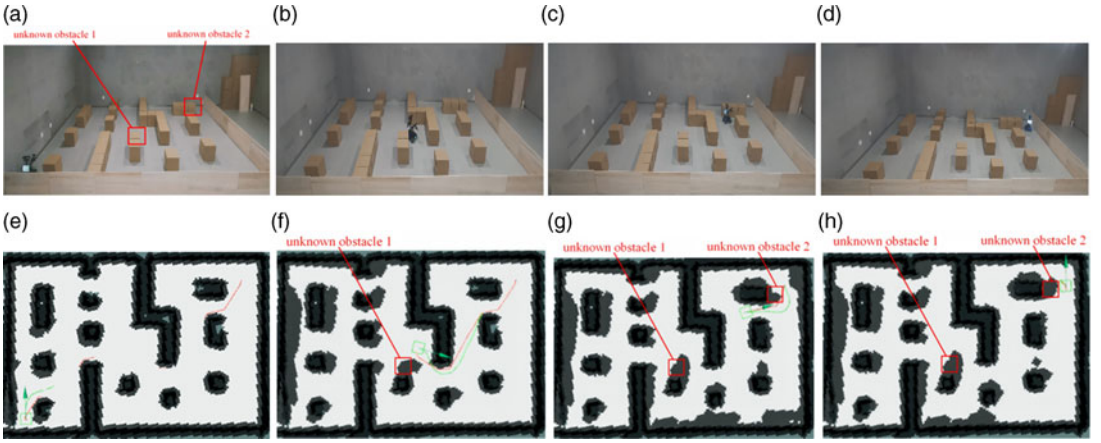


Figure 19. *Dynamic obstacle avoidance of unknown obstacles and their Rviz in ROS.*

In the dynamic path planning experiment, the robot's ability to avoid unknown obstacles during motion was evaluated. To simulate newly added unknown obstacles, two cardboard boxes were placed in the experimental environment, as shown in Figure 19. During the static path planning phase, when an unknown obstacle entered the detection range of the local path planning algorithm, the robot promptly detected and responded to the obstacle using the local dynamic fusion algorithm.

To evaluate the robot's dynamic obstacle avoidance capability, a dedicated experiment involving a moving obstacle was conducted separately in this study. Due to spatial constraints of the original experimental setup, a new test environment was constructed. In this experiment, the moving obstacle was represented by a human, traveling in the direction indicated by the red arrow at a speed ranging from 1 to 1.4 m/s. Figure 20 provides a detailed illustration of the robot's dynamic obstacle avoidance process. Additionally, Figure 21 depicts the path replanning procedure executed by the robot when encountering the moving obstacle during the experiment.

As shown in Figure 21, the dashed box represents the dynamic trajectory of the moving obstacle. The path replanning process during dynamic obstacle avoidance is divided into three time steps: t_1, t_2, and t_3. At time t_1, the moving obstacle begins its motion; at time t_2, the obstacle is in the process of moving; and at time t_3, the obstacle reaches its target position. During each movement of the obstacle, the robot replans its path based on environmental information, ensuring that each replanned path is optimal. This approach reduces redundant recalculations caused by the DWA during the obstacle avoidance process, demonstrating the strong path replanning capability of the ESSA-DWA hybrid algorithm.

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

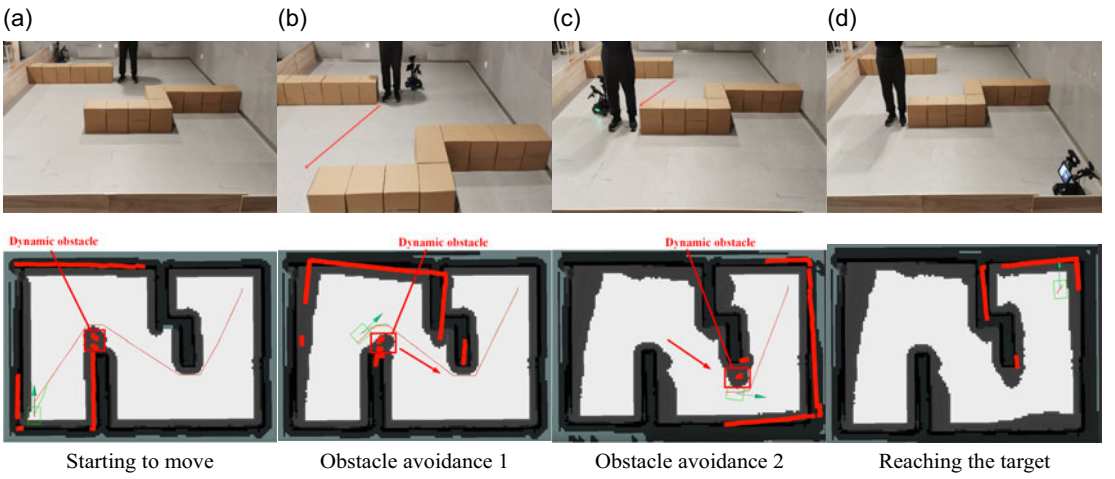


Figure 20. *Dynamic obstacle avoidance of moving obstacle and their Rviz in ROS.*

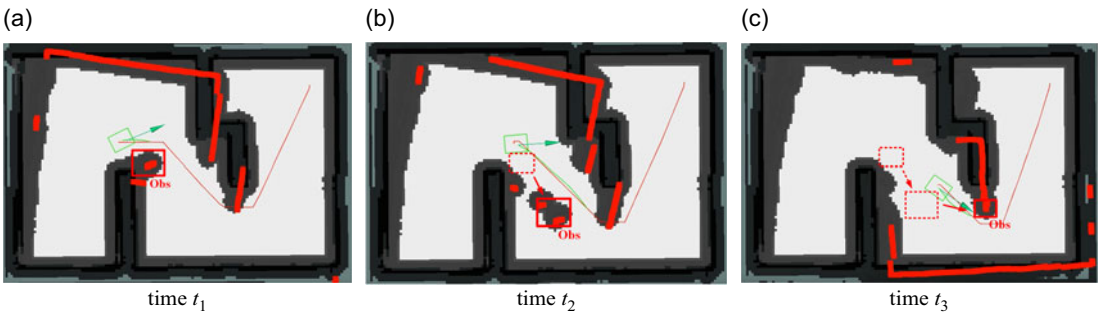


Figure 21. *Path replanning process of dynamic obstacle avoidance.*

As shown in Figures 19–21, the ESSA algorithm first plans a global path. As the robot moves, its LiDAR continuously scans the environment, enabling real-time perception and response to changes in obstacle positions while progressing toward the target. When approaching unknown obstacles, the robot rapidly performs secondary path planning based on updated environmental information. Each path generated during this process is optimal, relying on the DWA algorithm to avoid obstacles and achieve dynamic obstacle avoidance. Ultimately, the robot successfully reaches its target position.

The experimental results demonstrate that, whether in static or dynamic environments, the mobile robot can still plan its path reasonably using this algorithm and effectively bypass both static and unknown obstacles. This ensures the safe arrival of the robot at the target position, achieving the functions of path planning, autonomous navigation, and obstacle avoidance. These results further validate the effectiveness of the proposed algorithm.

6. Conclusions

This study proposes a path planning solution that integrates an ESSA with the DWA, effectively addressing the challenges faced by mobile robots in complex environments. Traditional SSA often suffers from issues such as susceptibility to local optima and excessively long path lengths during path planning, while dynamic obstacle avoidance remains a critical challenge for mobile robots. The proposed algorithm first employs ESSA to plan a global path, which then guides the DWA in planning local paths.

Firstly, ESSA utilizes Tent chaotic initialization to enhance population diversity, thereby avoiding premature convergence to local optima. Secondly, an adaptive adjustment of dynamic inertia weight is

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

implemented during the search process, effectively balancing the exploration and exploitation capabilities of the algorithm. Subsequently, a local search strategy is introduced to improve the precision of individual updates, further enhancing local search performance. Additionally, the Floyd path smoothing algorithm is applied to optimize the generated path, resulting in smoother trajectories. Finally, the integration of ESSA and DWA is achieved by extracting key nodes from the global path planned by ESSA to serve as references for DWA's local path planning. This ensures that the local path aligns more closely with the global path while enabling real-time obstacle detection and dynamic avoidance. For global path planning, we conducted three sets of simulation experiments on grid maps of sizes 20×20, 30×30, and 50×50, respectively. These experiments effectively validated the superior performance of the improved algorithm in various complex environments. For local path planning, simulations were performed using environment models 3, 6, and 9. The results demonstrated that the proposed algorithm exhibits remarkable performance across diverse complex scenarios, fully verifying its practicality and theoretical effectiveness.

Practical experiments further confirmed the significant advantages of this integrated algorithm in handling both static and dynamic obstacle avoidance tasks, showcasing its efficiency and reliability in dynamic environments. This approach provides a more effective and feasible solution for mobile robot path planning. However, there are areas for improvement in this study. At the algorithmic level, further refinements to the algorithm's strategies and structure could reduce time complexity and enhance operational efficiency. Additionally, improvements to the DWA algorithm could be explored, particularly in investigating the impact of dynamic obstacles on the robot's trajectory. At the application level, extending this algorithm to multirobot path planning problems could be a promising direction for future research.

Author contributions. Conceptualization, Y.L.; methodology, Y.L.; software, Y.L.; validation, Y.W. and K.L.; investigation, Y.L.; resources, S.L.; data curation, Y.L., Y.W. and K.L.; writing – original draft preparation, Y.L.; writing – review and editing, Y.L.; project administration, Y.L., S.L., Y.W. and K.L.; supervision and funding acquisition, S.L. All authors have read and agreed to the published version of the manuscript.

Financial support. This research was supported by the National Natural Science Foundation of China [Grant No. 51905364].

Competing interests. The authors declare no conflict of interest.

Ethical approval. The authors declare that this work is original and does not include experiments with animals.

References

[1] Y. Xu, Q. Li, X. Xu, J. Yang and Y. Chen, "Research progress of nature-inspired metaheuristic algorithms in mobile robot path planning," *Electronics* **12**(15), 3263 (2023). doi: 10.3390/electronics12153263.

[2] L. Liu, X. Wang, X. Yang, H. Liu, J. Li and P. Wang, "Path planning techniques for mobile robots: Review and prospect," *Expert Syst. Appl.* **227**, 120254 (2023). doi:10.1016/j.eswa.2023.120254.

[3] J. Zhang and J. Bai, "Dijkstra algorithm based cooperative caching strategy for UAV-assisted edge computing system," *Wirel Netw.* **30**(3), 1201–1219 (2024). doi:10.1007/s11276-023-03551-x.

[4] H. Zhang, Y. Tao and W. Zhu, "Global path planning of unmanned surface vehicle based on improved A-star algorithm," *Ah S Sens.* **23**(14), 6647 (2023). doi:10.3390/s23146647.

[5] K. Hao, Y. Yang, Z. Li, Y. Liu and X. Zhao, "CERRT: A mobile robot path planning algorithm based on RRT in complex environments," *Appl. Sci.* **13**(17), 9666 (2023). doi:10.3390/app13179666.

[6] C. Liu, S. Xie, X. Sui, Y. Huang, X. Ma, N. Guo and F. Yang, "PRM-D* method for mobile robot path planning," *Ah S Sens.* **23**(7), 3512 (2023). doi:10.3390/s23073512.

[7] Y. Yang, X. Luo, W. Li, C. Liu, Q. Ye and P. Liang, "AAPF*: A safer autonomous vehicle path planning algorithm based on the improved A* algorithm and APF algorithm," *Cluster Comput.*, 1–4 (2024).

[8] X. Yin, P. Cai, K. Zhao, Y. Zhang, Q. Zhou and D. Yao, "Dynamic path planning of AGV based on kinematical constraint A* algorithm and following DWA fusion algorithms," *Ah S Sens.* **23**(8), 4102 (2023). doi:10.3390/s23084102.

[9] A. V. Savkin and M. Hoy, "Reactive and the shortest path navigation of a wheeled mobile robot in cluttered environments," *Robotica* **31**(2), 323–330 (2013).

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press

[10] P. Zhou, Z. Xie, W. Zhou and Z. Tan, "A heuristic integrated scheduling algorithm based on improved Dijkstra algorithm," *Electronics* **12**(20), 4189 (2023). doi: 10.3390/electronics12204189.
[11] L. S. Jabbar, E. I. Abass and S. D. Hasan, "A modification of shortest path algorithm according to adjustable weights based on Dijkstra algorithm," *Eng. Technol. J.* **41**(2), 359–374 (2023).
[12] G. Tang, C. Tang, C. Claramunt, X. Hu and P. Zhou, "Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment," *IEEE Access* **9**, 59196–59210 (2021). doi:10.1109/ACCESS.2021.3070054.
[13] P. Li, L. Hao, Y. Zhao and J. Lu, "Robot obstacle avoidance optimization by A* and DWA fusion algorithm," *PLoS ONE* **19**(4), e0302026 (2024). doi:10.1371/journal.pone.0302026.
[14] T. V. Dang and P. X. Tan, "Hybrid mobile robot path planning using safe JBS-A* B algorithm and improved DWA based on monocular camera," *J. Intell. Robot. Syst.* **110**(4), 151 (2024). doi:10.1007/s10846-024-02179-z.
[15] Y. M. Liang and H. Y. Zhao, "CCPF-RRT*: An improved path planning algorithm with consideration of congestion," *Expert Syst. Appl.* **228**, 120403 (2023). doi:10.1016/j.eswa.2023.120403.
[16] M. Novosad, R. Penicka and V. Vonasek, "CTopPRM: Clustering topological PRM for planning multiple distinct paths in 3D environments," *IEEE Robot. Autom. Lett.* **8**(11), 7336–7343 (2023). doi:10.1109/LRA.2023.3315539.
[17] W. Zhang, N. Wang and W. Wu, "A hybrid path planning algorithm considering AUV dynamic constraints based on improved A* algorithm and APF algorithm," *Ocean Eng.* **285**, 115333 (2023). doi:10.1016/j.oceaneng.2023.115333.
[18] M. Li, B. Li, Z. Qi, J. Li and J. Wu, "Optimized APF-ACO algorithm for ship collision avoidance and path planning," *J. Mar. Sci. Eng.* **11**(6), 1177 (2023). doi:10.3390/jmse11061177.
[19] Z. Pan, M. Yuan, R. Wang, J. Wen, Q. Bi, J. Yuan and X. Zhang, "Dynamic' DWA for motion planning of mobile robots in dynamic environments," *Int. J. Dyn. Control* **11**(6), 3136–3144 (2023). doi:10.1007/s40435-022-01092-3.
[20] J. H. Holland, "Genetic algorithms, sci," *Am* **267**(1), 66–73 (1992).
[21] J. Kennedy and R. C. Eberhart, "A Discrete Binary Version of the Particle Swarm Algorithm," 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, vol. **5** (IEEE, 1997) pp. 4104–4108.
[22] M. Dorigo, M. Birattari and T. Stutzle, "Ant colony optimization," *IEEE Comput. Intell. Mag.* **1**(4), 28–39 (2007).
[23] S. Mirjalili, S. M. Mirjalili, A. Lewis and G. wolf optimizer, *Adv. Eng. Softw.* **69**, 46–61 (2014).
[24] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.* **95**, 51–67 (2016).
[25] H. Duan and P. Qiao, "Pigeon-inspired optimization: A new swarm intelligence optimizer for air robot path planning," *Int. J. Intell. Comput. Cybern.* **7**(1), 24–37 (2014).
[26] Y. Li, J. Zhao, Z. Chen, G. Xiong and S. Liu, "A robot path planning method based on improved genetic algorithm and improved dynamic window approach," *Sustainability-BASEL* **15**(5), 4656 (2023). doi:10.3390/su15054656.
[27] S. Lin, A. Liu, J. Wang and X. Kong, "An intelligence-based hybrid PSO-SA for mobile robot path planning in warehouse," *J. Comput. Sci.* **67**, 101938 (2023). doi:10.1016/j.jocs.2022.101938.
[28] T. Zhou and W. Wei, "Mobile robot path planning based on an improved ACO algorithm and path optimization," *Multimed. Tools Appl.*, 1–24 (2024). doi:10.1007/s11042-024-19370-x.
[29] X. Yu, N. Jiang, X. Wang and M. Li, "A hybrid algorithm based on grey wolf optimizer and differential evolution for UAV path planning," *Expert Syst. Appl.* **215**, 119327 (2023). doi:10.1016/j.eswa.2022.119327.
[30] Y. Dai, J. Yu, C. Zhang, B. Zhan and X. Zheng, "A novel whale optimization algorithm of path planning strategy for mobile robots," *Appl. Intell.* **53**(9), 10843–10857 (2023). doi:10.1007/s10489-022-04030-0.
[31] Y. Jiang, T. Bai, D. Wang and Y. Wang, "Coverage path planning of UAV based on linear programming—fuzzy C-means with pigeon-inspired optimization," *Drones* **8**(2), 50 (2024). doi:10.3390/drones8020050.
[32] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: Sparrow search algorithm," *Syst. Sci. Control Eng.* **8**(1), 22–34 (2020).
[33] Y. Zheng, L. Li, L. Qian, B. Cheng, W. Hou and Y. Zhuang, "Sine-SSA-BP ship trajectory prediction based on chaotic mapping improved sparrow search algorithm," *Ah S Sens.* **23**(2), 704 (2023). doi:10.3390/s23020704.
[34] A. M. Khedr, Z. A. Aghbari and P. P. Raj, "MSSPP: Modified sparrow search algorithm based mobile sink path planning for WSNs," *Neural Comput. Appl.* **35**(2), 1363–1378 (2023). doi:10.1007/s00521-022-07794-1.
[35] G. Liu, S. Zhang, G. Ma and Y. Pan, "Path planning of unmanned surface vehicle based on improved sparrow search algorithm," *J. Mar. Sci. Eng.* **11**(12), 2292 (2023). doi:10.3390/jmse11122292.
[36] Z. Wang, G. Sun, K. Zhou and L. Zhu, "A parallel particle swarm optimization and enhanced sparrow search algorithm for unmanned aerial vehicle path planning," *Heliyon* **9**(4), e14784 (2023). doi:10.1016/j.heliyon.2023.e14784.
[37] L. Liu, J. Liang, K. Guo, C. Ke, D. He and J. Chen, "Dynamic path planning of mobile robot based on improved sparrow search algorithm," *Biomimetics* **8**(2), 182 (2023). doi:10.3390/biomimetics8020182.
[38] J. Hou, W. Jiang, Z. Luo, L. Yang, X. Hu and B. Guo, "Dynamic path planning for mobile robots by integrating improved sparrow search algorithm and dynamic window approach," *Actuators* **13**(1), 24 (2024). doi:10.3390/act13010024.

Cite this article: Y. Luo, S. Lin, Y. Wang and K. Liang (2025). "Path planning for mobile robots in complex environments based on enhanced sparrow search algorithm and dynamic window approach", Robotica **43**, 1929–1952. https://doi.org/10.1017/S026357472500061X

https://doi.org/10.1017/S026357472500061X Published online by Cambridge University Press