actuators

MDPI

Article

Dynamic Path Planning for Mobile Robots by Integrating Improved Sparrow Search Algorithm and Dynamic Window Approach

Junting Hou [1], Wensong Jiang [1], Zai Luo [1,*], Li Yang [2], Xiaofeng Hu [1] and Bin Guo [1]

[1] College of Metrology and Measurement Engineering, China Jiliang University, Hangzhou 310018, China; houjt@cjlu.edu.cn (J.H.); jwensong@cjlu.edu.cn (W.J.); 12b0202094@cjlu.edu.cn (X.H.); guobin905@cjlu.edu.cn (B.G.)

[2] College of Information Engineering, China Jiliang University, Hangzhou 310018, China; lyang@cjlu.edu.cn

* Correspondence: luozai@cjlu.edu.cn

Abstract: To overcome the limitations of the sparrow search algorithm and the challenges of dynamic obstacle avoidance in mobile robots, an integrated method combining the enhanced sparrow search algorithm with the dynamic window approach is introduced. First, logistic–tent chaotic mapping is utilized for the initialization of the sparrow population, thereby achieving a uniform distribution of the sparrow population and simultaneously enhancing the exploratory capability of the algorithm. The implementation of the elite reverse learning strategy aims to diversify the sparrow population, thus improving the quality of initial solutions and the algorithm's search accuracy. Additionally, the position update dynamic self-adaptive adjustment strategy is adopted to enhance the optimization capability of the algorithm by refining the position update formulas for both producers and scroungers. By combining the Lévy flight strategy and the optimal position perturbation strategy, the algorithm's efficacy in escaping local optima can be improved. Second, an adaptive velocity adjustment strategy is presented for the dynamic window approach and optimized for its evaluation function to enhance the safety of the path. Third, the enhanced sparrow search algorithm is integrated with the dynamic window approach to tackle the problems of the non-smooth global path and inadequate dynamic obstacle avoidance capability. Both simulation and experimental results show the superiority of the enhanced sparrow search algorithm in comparison to other algorithms in terms of the path length, total rotation angle, and algorithm execution time. Notably, in comparison to the basic sparrow search algorithm, there is a decrease in average path lengths by 15.31% and 11.92% in the improved sparrow search algorithm. The integrated algorithm not only crafts local paths rooted in global paths but also adeptly facilitates real-time dynamic obstacle evasion, ensuring the robot's safe arrival at its destination.

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

check for updates

Citation: Hou, J.; Jiang, W.; Luo, Z.; Yang, L.; Hu, X.; Guo, B. Dynamic Path Planning for Mobile Robots by Integrating Improved Sparrow Search Algorithm and Dynamic Window Approach. *Actuators* **2024**, *13*, 24. https://doi.org/10.3390/act13010024

Academic Editors: Myun Joong Hwang and Maolin Jin

Received: 5 December 2023
Revised: 1 January 2024
Accepted: 6 January 2024
Published: 8 January 2024

1. Introduction

In recent years, the rapid growth of the artificial intelligence field has led to extensive applications of mobile robots in various areas, including transportation, rescue operations, military applications, and agriculture [1–3]. In mobile robot research, a core topic is path planning. Essentially, a robot must autonomously devise a safe, collision-free route from its starting point to its destination, considering the known environment. An efficient path-planning algorithm, characterized by quick optimization, concise routes, and strong obstacle avoidance, not only boosts a mobile robot's task efficiency but also guarantees safe navigation [4–6].

Mobile robot path planning can be classified into global path planning, relying on pre-existing knowledge, and local path planning, utilizing sensor-derived information, depending on the comprehension of environmental data [7]. Global path planning involves charting an optimal path for a mobile robot in a completely known environment. Predominantly, traditional algorithms for global path planning encompass Dijkstra [8], the A* algorithm [9], and rapidly exploring random trees (RRTs) [10]. Local path planning, also known as dynamic path planning, places a stronger emphasis on the real-time dynamic information of the robot's current environment [11]. It utilizes sensors to perform real-time detection of the surroundings, enabling the robot to achieve dynamic obstacle avoidance. Local path planning algorithms primarily include the dynamic window approach (DWA) [12], artificial potential field (APF) [13], and the time elastic band (TEB) [14]. Drawing inspiration from nature, a surge in intelligent optimization algorithms has emerged for robot path planning. Notable ones include the genetic algorithm (GA) [4], ant colony optimization (ACO) [15], particle swarm optimization (PSO) [16], grey wolf optimization (GWO) [17], and the artificial bee colony algorithm (ABC) [18]. Numerous scholars have conducted extensive research to improve and optimize the aforementioned algorithms, thereby enhancing their performance. Zheng et al. [19] utilized the actual shortest distance traversed by ants to update heuristic information and introduced reward and penalty rules to optimize the local pheromone update strategy, thus enhancing the convergence speed and optimal solution capability of the ant colony algorithm. Raj et al. [20] addressed the issue of autonomous mobile robots collecting information in unknown areas with energy constraints and time-sensitive preferences. They proposed an intelligent PSO technique, which determines fitness values by simplifying optimization tasks and improves the velocity update formula for each particle. This effectively optimizes the energy consumption and time constraints of mobile robot navigation. Dong et al. [21] utilized three improvement strategies to balance the global and local search capabilities of the GWO. This approach mitigates the algorithm's slow convergence and vulnerability to local optima issues to some extent. Eshtehardian et al. [22] proposed a method that combines RRT* and B-spline optimization to smooth the paths generated by the RRT* algorithm. They also incorporated some optimization functions into the RRT* algorithm and provided correction methods, validating the effectiveness of the algorithm in the Webots simulation environment. Zhang et al. [23] proposed a path-planning approach that combines ant colony algorithms with the A* shortest-path search algorithm. They optimized the heuristic function, enabling the method to swiftly and effectively obtain an optimal path.

Drawing inspiration from the collective intelligence, foraging, and anti-predator behaviors of sparrows, Xue et al. [24] introduced a novel swarm intelligence optimization algorithm in 2020, termed the sparrow search algorithm (SSA). By simulating the foraging behavior of sparrow populations, this algorithm optimizes the exploration and exploitation capabilities of the search space to some extent. It showcases robust competitiveness concerning both convergence speed and stability when compared to alternative algorithms. However, during the later stages of the algorithm's search for global optima, issues such as reduced population diversity, insufficient local exploration, and vulnerability to local optima arise [25]. Liu et al. [26] achieved a balance between the exploitation and exploration capabilities of the SSA using adaptive weight factors. They utilized the Cauchy Gaussian mechanism to enhance SSA's ability to break free from stagnation, thereby verifying the superiority of this enhancement strategy. Zhou et al. [27] proposed the incorporation of genetic algorithm crossover and mutation principles to enhance the convergence capability of the algorithm. Khedr et al. [28] proposed a vehicle clustering algorithm that combines the GA and SSA to address the vehicle clustering problem in the Internet of Vehicles. This algorithm integrates the advantages of weight-based and mobility-based clustering methods, employing a fitness function that considers mobility metrics on clustering distance. This design results in superior performance in terms of cluster count and average cluster lifespan compared to the current state-of-the-art algorithms. Zhang et al. [29] employed a linear path strategy in conjunction with neighborhood search strategies to enhance the fitness value of the global optimal individuals. They utilized an improved position update function to accelerate convergence speed and elevate the quality of planned paths.

Geng et al. [30] improved the development and exploration capabilities of the SSA by introducing an adaptive weight-balancing algorithm. They also employed an adaptive spiral search method to enhance the performance of the SSA, resulting in improved stability and convergence accuracy of the algorithm. Khaleel [31] combined the SSA with differential evolution (DE), fully utilizing the advantages of these two heuristic algorithms. This integration improved the efficiency of job scheduling on a heterogeneous cloud computing platform, reducing system response time and energy consumption and effectively enhancing resource utilization.

The above research optimized the algorithm in terms of efficiency, computational complexity, energy efficiency, etc. The use of various optimization strategies and the fusion of multiple algorithms have certain reference significance for subsequent research, improving the performance of the algorithm to a certain extent and enhancing the efficiency and safety of path planning. Although the aforementioned studies on the SSA somewhat improved the algorithm's performance, there is still room for optimization and improvement. Moreover, the current research on the SSA predominantly focuses on algorithmic theory and the ability to solve objective functions. Research applying the SSA to the field of robot path planning remains relatively limited, and there is a scarcity of studies investigating the algorithm's applicability to dynamic path-planning challenges in complex environments for robots. Considering that mobile robots predominantly operate in dynamic work environments, the efficient planning of a globally optimal path while ensuring dynamic obstacle avoidance holds research significance and practical value. Through analyzing previous research, it is known that global path-planning algorithms can easily obtain the optimal path, but they cannot avoid unknown static and dynamic obstacles that may appear during the planning process. Local path-planning algorithms can navigate obstacles encountered during travel, but they are prone to falling into local optima [32,33].

Building on prior research, this paper addresses the challenges of mobile robot path planning and dynamic obstacle avoidance by proposing an integrated approach that combines the improved SSA with the DWA for path planning. Initially, addressing concerns arising during the iterative process of the basic SSA, such as insufficient population diversity, inadequate local exploration, slowed convergence speed, and poor ability to escape local optima, a multi-strategy optimization approach is integrated to enhance algorithm performance. This gives rise to the multi-strategy improved sparrow search algorithm (MISSA), facilitating the attainment of a globally optimal path. Subsequently, within the DWA, an adaptive velocity adjustment strategy is incorporated to enhance safety during the motion process. Furthermore, the evaluation function of the DWA is optimized and adjusted to improve its dynamic obstacle avoidance capability. Finally, the integration of the aforementioned two algorithms involves using the globally significant waypoints generated by the improved SSA as local subgoals within the DWA. This facilitates localized dynamic path planning through the application of the DWA. By ensuring that the globally planned path remains optimal, this integrated approach achieves dynamic obstacle avoidance for unknown obstacles, thereby yielding a smooth path for real-time obstacle avoidance and enhancing the overall efficiency of robot movement.

The remaining sections of this paper are organized as follows: Section 2 introduces the basic SSA and the SSA with multiple strategy improvements. Section 3 discusses the improved DWA and the MISSA-DWA fusion algorithm of dynamic path planning. Section 4 analyzes simulation and experimental results, confirming the superiority of the improved algorithm and the effectiveness of the fusion algorithm through comparative experiments. Finally, Section 5 gives the conclusion and outlines the next steps in the research plan.

2. Global Path Planning

2.1. Basic SSA

The SSA is an inspired heuristic algorithm derived from the predatory and anti-predatory behaviors observed in sparrows in the biological world. The algorithm's construction is based on the behavior of sparrows, categorizing the sparrow population into producers, scroungers, and a certain number of randomly generated scouters [24]. The matrix for the sparrow population is defined as

$$\boldsymbol{X} = [x_1, x_2, \cdots, x_N]^{\mathrm{T}},\ x_i = [x_{i,1}, x_{i,2}, \cdots, x_{i,d}] \tag{1}$$

where N is the size of the sparrow population, which corresponds to the number of sparrows in the population $i = 1, 2, \cdots, N$ and represents the dimension of the variables in the optimization problem.

The matrix representing the fitness values of sparrows is as follows:

$$\boldsymbol{F}_x = [\boldsymbol{f}(x_1), \boldsymbol{f}(x_2), \cdots, \boldsymbol{f}(x_n)]^{\mathrm{T}} \tag{2}$$

$$f(x_i) = [f(x_{i,1}), f(x_{i,2}), \cdots, f(x_{i,d})] \tag{3}$$

where n represents the number of sparrows, while each value in $\boldsymbol{F}_x$ represents the fitness value of an individual. Sparrows that possess higher fitness values are the ones securing food first, and as pioneers, they lead the entire population toward the food source, guiding the direction of the population's movement [34]. The updating mechanism for the positions of the producers is as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{\max}}\right), R_2 < ST \\ X_{i,j}^t + Q \cdot \boldsymbol{L}, R_2 \geqslant ST \end{cases} \tag{4}$$

where $X_{i,j}^{t+1}$ is the position of the i-th sparrow in the j-th dimension, $j = 1, 2, \cdots, d$, and t represents the current iteration count. $iter_{\max}$ stands for the maximum number of iterations, $\alpha \in (0, 1)$ is a random number within the range, $R_2(R_2 \in [0, 1])$ and $ST(ST \in [0.5, 1.0])$ respectively denote the warning and safety values. Q is a random number drawn from a normal distribution [0, 1]. $\boldsymbol{L}$ is a matrix of $1 \times d$, with each element in the matrix being 1. When $R_2 < ST$, it indicates the absence of predators in the vicinity, and producers employ an extensive search pattern. If $R_2 \geq ST$, it suggests that some members have detected predators, necessitating the entire population to swiftly relocate to safer regions.

Except for the group of producers, individuals within the rest of the sparrow populations are considered scroungers, who update their positions by following the producers and foraging around them [35]. The formula for updating the scrounger's position is defined as

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{i,j}^t}{i^2}\right), i > \frac{N}{2} \\ X_p^{t+1} + \mid X_{i,j}^t - X_p^{t+1} \mid \cdot \boldsymbol{A}^+ \cdot L, i \leqslant \frac{N}{2} \end{cases} \tag{5}$$

where X_{worst} represents the globally worst position. $\boldsymbol{A}$ is a matrix of $1 \times d$ with each element assigned a value of 1 or -1 randomly. $\boldsymbol{A}^+ = \boldsymbol{A}^{\mathrm{T}}(\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}})^{-1}$. When $i > N/2$, it indicates that the i-th scrounger with a poorer fitness value has failed to secure food, possesses diminished energy levels, and hence needs to venture into other regions in search of nourishment to replenish energy.

During foraging in the sparrow population, a subset of sparrows is chosen to assume vigilance responsibilities, guiding the group to evade predators and ensuring the safety of the entire population. Each generation randomly selects a certain number (typically 10%~20%) of scouters from the population to engage in vigilant behaviors [24]. The formula for updating the positions of scouters is given by

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^t + \beta \cdot \mid X_{i,j}^t - X_{\text{best}}^t \mid, f_i > f_g \\ X_{i,j}^t + k \cdot \frac{|X_{i,j}^t - X_{\text{worst}}^t|}{f_i - f_w + \varepsilon}, f_i = f_g \end{cases} \tag{6}$$

where X_{best} represents the optimal global position, β denotes the coefficient for adjusting step size, and $k \in [-1, 1]$ stands for a uniformly distributed random number within the specified range. ε is the minimum constant to prevent division by zero. f_i is the current sparrow's fitness value. f_g and f_w respectively indicate the present optimal and least advantageous fitness values at a global scale. When $f_i > f_g$, it suggests that the sparrow must adjust its position to move closer to the optimal fitness value, and when $f_i = f_g$, it signifies that this sparrow is moving toward a more favorable group of sparrows. k signifies the sparrow's movement direction and serves as the coefficient for adjusting step size.

2.2. MISSA: Sparrow Search Algorithm Improved through Integration of Multiple Strategies

During the iterative process, the SSA encounters challenges like a lack of population diversity, limited local exploration, reduced convergence speed, and a diminished capacity to break away from local optima. In response to these limitations, this paper introduces an SSA that integrates multiple strategies for improvement:

(1) To address the imbalanced distribution of the population and the inadequate diversity in the basic SSA, logistic–tent chaotic mapping is employed to initialize the sparrow population, ensuring an even distribution and improving algorithm traversal. In order to enhance the diversity within the population and improve both the quality of the initial solution and search precision, an elite-inverse learning strategy is implemented.

(2) To tackle the problem of inadequate position updating in the sparrow population, a dynamic self-adaptive adjustment strategy for position updating is employed. This strategy refines the position update equations for both producers and scroungers, bolstering the algorithm's optimization prowess.

(3) Given that the basic SSA tends to get trapped in local optima, a Lévy flight strategy is utilized to update the position of the scroungers. The incorporation of an optimal position perturbation strategy boosts the algorithm's capability to evade local optima.

2.2.1. Initialization of Population Using Logistic–Tent Chaotic Mapping

During the initialization stage, the random generation of sizes for various dimensions of the sparrow population leads to the issue of initial solutions clustering together. This results in low coverage and uneven distribution within the solution space, along with reduced diversity among individuals, subsequently affecting the optimization effectiveness and convergence efficiency of the algorithm. To address this problem, chaotic mapping is employed to establish a relatively uniform and stable initial population.

Chaotic mapping possesses high sensitivity and randomness, making it an effective tool for introducing randomness in search algorithms. Logistic chaotic mapping [36] and tent chaotic mapping [37] are currently common methods of chaotic mapping. Logistic chaotic mapping has a straightforward structure and is capable of generating highly intricate chaotic dynamic characteristics. However, when searching for optimal solutions, it is constrained by its uneven function distribution, with its values being larger on both ends and smaller in the middle. The definition formula for logistic chaotic mapping is defined as

$$x_{n+1} = \mu x_n (1 - x_n) \tag{7}$$

where x represents the system variable. k stands for the control parameter. $x \in [0, 1]$, and $\mu \in [3.57, 4]$.

Tent chaotic mapping is a piecewise linear one-dimensional mapping, possessing uniform probability density and power spectral density. However, it has issues such as a limited mapping range, small parameter space, and the existence of rational fixed points. It is defined as

$$x_{n+1} = \begin{cases} kx_n/2, x_n < 0.5 \\ k(1 - x_n)/2, x_n \geq 0.5 \end{cases} \tag{8}$$

where x represents the system variable, k denotes the control parameter, and $x \in [0, 1]$. When $k \in [2, 4]$, the system is in a chaotic state.

This paper employs the logistic–tent compound chaotic system generated by blending the logistic chaotic system and the tent chaotic system. This chaotic system integrates the intricate chaotic dynamic characteristics of the logistic chaotic system with the faster iteration speed, higher autocorrelation, and suitability for large datasets of the tent chaotic system. The logistic–tent compound chaotic system is employed in this paper for population initialization in the sparrow algorithm. Its mathematical expression is

$$X_{n+1} = \begin{cases} (rX_n(1-X_n) + (4-r)X_n/2) \bmod 1, X_n < 0.5 \\ (rX_n(1-X_n) + (4-r)(1-X_n)/2) \bmod 1, X_n \geq 0.5 \end{cases} \tag{9}$$

where X represents the system variable. r stands for the control parameter. $X \in [0,1]$, and $r \in [0,4]$. The distribution diagram of the logistic–tent chaotic system in the range [0, 1] is shown in Figure 1.

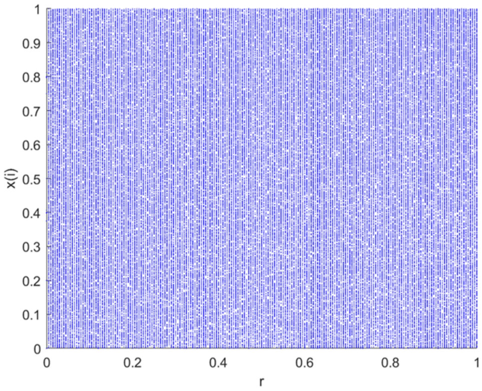


Figure 1. Distribution diagram of the logistic–tent compound chaotic mapping.

The trajectory distribution plot of the logistic–tent composite chaotic mapping, as shown in Figure 1, reveals that variables x and y in this chaotic mapping can access or gradually approach the entire data range. The trajectory is uniformly distributed across the entire phase plane, indicating the robust chaotic behavior of the logistic–tent composite chaotic mapping across the entire parameter range [38]. In comparison to logistic and tent chaotic mappings, the logistic–tent composite chaotic mapping exhibits more pronounced chaotic characteristics. The combination of the two enhances the complexity and randomness of the chaotic system, contributing to better coverage of the search space.

Utilizing the logistic–tent composite chaotic mapping for population initialization in the SSA effectively combines the randomness and complexity of chaotic systems, contributing to an improved global search capability. This aids in more effectively traversing the solution space and discovering the global optimal solution or better solutions.

2.2.2. Elite Opposition-Based Learning Strategy

The opposition-based learning strategy (OBL) [39] is a novel strategy proposed in the field of computational intelligence. Existing research has shown that, compared to the current solution, there is an approximate 50% likelihood that the inverse solution is in proximity to the global optimum. By employing this strategy, it is possible to effectively enhance the diversity of the population, thereby improving the global search efficiency of the algorithm.

The elite opposition-based learning strategy (EOBL) [40], in comparison to OBL, is more adept at finding global optimal solutions within the current search space. This strategy employs individuals to construct an oppositional population and simultaneously selects from the producer population within the sparrow population, enhancing the quality of the producer population and augmenting population diversity, thus laying the foundation for better global search outcomes.

If $X_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ is a regular particle with its corresponding self-best value being the elite particle $X_i^e = (x_{i1}^e, x_{i2}^e, \cdots, x_{iD}^e)$, then the elite oppositional solution can be defined as

$$X_i^e = \mu \times (da + db) - x_{ij}^e \tag{10}$$

where $x_{ij}^e \in [a_j, b_j]$. $\mu \in (0,1)$ represents a random number that adheres to a normal distribution. $[da_j, db_j]$ denotes the dynamic boundary of the j-dimensional search space, which can be computed using Equation (11).

$$\begin{cases} da_j = \min(x_{ij}) \\ db_j = \max(x_{ij}) \end{cases} \tag{11}$$

Utilizing dynamic boundaries in place of fixed boundaries for the search space allows for the accumulation of search experience, positioning the generated reverse solution within a progressively shrinking search space, thereby expediting the algorithm's convergence. If the generated oppositional solution lies outside the boundaries, it is reset using a method of random generation. This is illustrated by

$$X_i^e = \mathrm{rand}(da, db) \tag{12}$$

2.2.3. Dynamic Self-Adaptive Position Update Strategy

(1) Improved Formula for Producer Position Update

As indicated by Equation (5), during the predation process, the producer updates its position through a step size factor $\exp\left(\frac{-i}{\alpha \cdot iter_{\max}}\right)$. With the increase in iteration times, the step size factor decreases non-linearly, and the producer gradually establishes an encircling formation in this process. The advantage of this position update method is that it gradually narrows the search scope of the sparrow population, thereby enhancing the algorithm's local search capability. However, due to the excessive concentration of the population, the algorithm tends to lose its diversity in the later stages of iteration. This insufficient search range makes it prone to fall into local optima, thereby reducing the algorithm's global search precision. In the position update formula for producers, this paper incorporates the global optimal solution from the preceding generation. Thus, the updated position of the producer is influenced both by the position of the previous generation's producer and its global optimal solution. Adopting this strategy can effectively reduce the likelihood of the algorithm getting trapped in local optima. Additionally, we incorporate the sine-cosine optimization algorithm (SCA) [41] to modify the producer's position update formula, introducing a dynamic weight factor ω that starts with a larger value in the early iterations and adaptively decreases as iterations progress. Implementing this strategy ensures a more effective blend of global exploration and local search, ultimately boosting the algorithm's rate of convergence.

The formula for the dynamic weight factor ω and the improved producer's position update method are presented as

$$\omega = \frac{\mathrm{e}^{2(1-\sin(\frac{t\pi}{iter_{\max}}))} - \mathrm{e}^{-2(1-\sin(\frac{t\pi}{iter_{\max}}))}}{\mathrm{e}^{2(1-\sin(\frac{t\pi}{iter_{\max}}))} + \mathrm{e}^{-2(1-\sin(\frac{t\pi}{iter_{\max}}))}} \tag{13}$$

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t + \omega \sin(r_1) \left| r_2 \cdot X_{best} - X_{i,j}^t \right|, R_2 < ST \\ X_{i,j}^t + \omega \cos(r_1) \left| r_2 \cdot X_{best} - X_{i,j}^t \right|, R_2 \geqslant ST \end{cases} \tag{14}$$

where r_1 is a random number within the range of $[0, 2\pi]$, r_2 denotes a random number within the range of $[0, 2]$, and X_{best} represents the current global optimal position.

(2) Improved Formula for Scrounger Position Update

When the producer undergoes a certain number of iterations without changes in fitness value, the producer will transform into a producer. In order to avoid the algorithm from being ensnared in local optima, the Lévy flight strategy [42] is introduced into the producer position update formula to enhance the global search capability of the SSA. It is defined as

$$\text{Levy}(x) = 0.01 \cdot \frac{u \cdot \sigma}{|v|^{(1/\xi)}} \tag{15}$$

where u and v are standard normal distribution random numbers within the range of [0, 1], $\xi = 1.5$, and σ can be solved by

$$\sigma = \left(\frac{\Gamma(1+\xi) \cdot \sin(\pi\xi/2)}{\Gamma((1+\xi)/2) \cdot \xi \cdot 2^{((\xi-1)/2)}} \right)^{(1/\xi)} \tag{16}$$

where $\Gamma(x)$ is the gamma function, which is denoted as $\Gamma(x) = (x-1)!$.

The improved formula for producer position update is as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^{t} - X_{i,j}^{t}}{i^2} \right), i > \frac{N}{2} \\ X_p^{t+1} + X_p^{t+1} \otimes \text{Levy}(d), i \leqslant \frac{N}{2} \end{cases} \tag{17}$$

where X_p^{t+1} is the current optimal position of the producer.

2.2.4. Optimal Position Perturbation Strategy

In the later iterations of the SSA, the sparrow population progressively approaches optimal individuals, causing a reduction in population diversity and rendering the algorithm susceptible to local optima. To tackle this problem, this study introduces a Cauchy-t perturbation strategy to disturb the top-ranked sparrow individuals. By comparing positions before and after perturbation, superior positions are selected to enter the subsequent iteration, thereby enhancing the algorithm's capability to break free from local optima.

Inspired by reference [26], the Cauchy-t perturbation operator is constructed, with its expression given by

$$X_{i,j}^{t+1} = X_{\text{best}}^{t}(1 + \lambda_1 \text{Cauchy}(0,1)) + \lambda_2 t(T_{\text{max}}) \tag{18}$$

where $\text{Cauchy}(0,1)$ is a random number following the standard Cauchy distribution, and $t(T_{\text{max}})$ stands for a t-distributed random number with its degrees of freedom determined by the maximum number of iterations. The Cauchy distribution operator possesses strong perturbation capabilities, contributing to the enhanced global search performance of the algorithm. The t-distribution exhibits similar characteristics to the Cauchy distribution when its degrees of freedom are low, showcasing robust perturbation capabilities. However, when the degrees of freedom are high, it demonstrates Gaussian distribution characteristics, which can enhance the algorithm's local search capability.

In Equation (19), the combination of random numbers following the Cauchy distribution and t-distribution is achieved through λ_1 and λ_2, where $\lambda_1 = 1 - t^2/T_{max}^2$ and $\lambda_2 = t^2/T_{max}^2$ are adaptive dynamic parameters varying with the number of iterations. During the initial stages of the algorithm, the higher value of λ_1 indicates a more significant influence of the Cauchy perturbation, allowing for perturbations of the optimal sparrow position within a larger search range to enhance the algorithm's global search capability. In the subsequent iterations, the larger value of λ_2 implies a more pronounced effect of the t-distribution perturbation, facilitating exploration near the optimal sparrow position and enhancing the algorithm's local exploitation capability, thus enhancing the convergence accuracy of the SSA.

2.2.5. Improved Algorithm Flow

The specific implementation process of the MISSA is as follows:

Step 1: Initialize parameters. These include population size, maximum number of iterations, proportion of producers, proportion of scouters, warning threshold, and safety threshold.
Step 2: Initialize the population through the application of logistic–tent chaotic mapping, calculate the fitness value for each sparrow, sort them, and identify the current best and worst fitness values along with their corresponding sparrow positions.
Step 3: Apply the EOBL strategy proportionate to the number of producers, combined with fitness value sorting, to select top-ranked sparrows as producers and update their positions according to Equations (13) and (14).
Step 4: The remaining sparrows function as producers and update their positions using Equation (17).
Step 5: Based on the proportion of scouters, randomly select scouters from the sparrow population and update their positions by using Equation (5).
Step 6: Compute the fitness value for each sparrow and sort them. When the sparrow individuals gather to a certain extent, apply the Cauchy perturbation strategy to disturb the optimal sparrow position.
Step 7: Compare the newly perturbed fitness value with the original value and update individual positions accordingly.
Step 8: Determine whether the maximum iteration count has been reached. If true, end the loop, output, and record the optimal result. If not, proceed to step 3.
Step 9: Output the global optimal path and its corresponding fitness value.

3. Integration of Enhanced DWA for Dynamic Path Planning

MISSA has optimized and improved upon the issues present in the basic SSA. By specifying the starting point and destination on the grid map, MISSA can produce an optimal global path from the initial point to the endpoint. However, due to the absence of dynamic obstacle information in MISSA, real-time dynamic obstacle avoidance cannot be performed. To address this issue, this paper integrates MISSA with the DWA algorithm. Once MISSA has planned a global path, a local motion strategy for the robot is developed based on global path information and real-time obstacle data, achieving both global path optimization and local dynamic obstacle avoidance for the mobile robot.

3.1. Improvements to the DWA

The DWA involves sampling multiple sets of velocity pairs in the velocity space, simulating the robot's trajectory under these velocity spaces, and then using an evaluation function to assess the simulated trajectories. The optimal trajectory and corresponding velocity pair are selected from these evaluations to achieve local dynamic path planning [43]. Building upon the robot's motion model and velocity sampling, this paper enhances the evaluation function of the DWA by incorporating a velocity-adaptive adjustment strategy.

3.1.1. Establishment of Robot Kinematic Model

To perform velocity and motion trajectory simulation in the DWA algorithm, it is essential to establish a kinematic model. The motion model for the robot to perform uniform motion within time Δt is as follows:

$$\begin{cases} x_{t+1} = x_t + v_x \Delta t \cos(\theta_t) - v_y \Delta t \sin(\theta_t) \\ y_{t+1} = y_t + v_x \Delta t \sin(\theta_t) + v_y \Delta t \cos(\theta_t) \\ \theta_{t+1} = \theta(t) + \omega_t \Delta(t) \end{cases} \quad (19)$$

The kinematic model of the robot is illustrated in Figure 2.

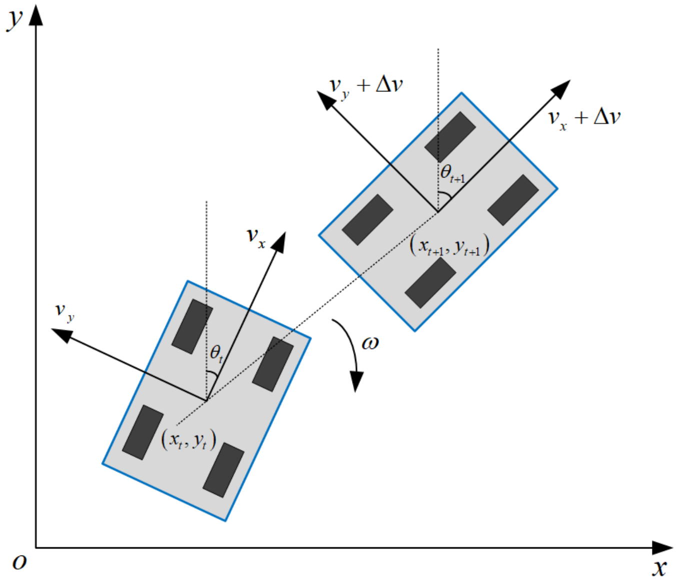


Figure 2. Robot kinematic model.

3.1.2. Velocity Sampling

After establishing the robot's kinematic model, its motion trajectory is predicted through velocity sampling. During velocity sampling, multiple sets of sampled velocities are usually obtained to create a dynamic window of velocities. These sampled velocities are evaluated using a cost function to determine the relatively optimal velocities at different moments. Based on this evaluation, the robot's motion trajectory is projected. Within the velocity space (v, m), multiple sets of velocity pairs exist, and due to the robot's inherent characteristics and environmental constraints, velocities are restricted within a certain range.

(1) Robot's Maximum and Minimum Velocities:

$$V_m = \{(v, \omega) | v \in [v_{\min}, v_{\max}], \omega \in [\omega_{\min}, \omega_{\max}]\} \tag{20}$$

where $v_{\min}$ and $v_{\max}$ represent the robot's maximum and minimum linear velocities, and $\omega_{\min}$ and $\omega_{\max}$ denote the robot's maximum and minimum angular velocities.

(2) Robot's Intrinsic Constraints

Considering the constraints imposed by actual motor torque, the robot is subject to maximum acceleration and deceleration. The velocity constraint formula is provided in Equation (21).

$$V_d = \left\{ \begin{array}{l} (v, \omega) \mid v \in [v_c - \dot{v}_b \Delta t, v_c + \dot{v}_a \Delta t] \\ \quad \omega \in [\omega_c - \dot{\omega}_b \Delta t, \omega_c + \dot{\omega}_a \Delta t] \end{array} \right\} \tag{21}$$

where V_d represents the velocity space achievable by the robot, v_c and ω_c are the linear and angular velocities at the current moment, and $\dot{v}_b$ and $\dot{v}_a$ denote the maximum acceleration of the robot. $\dot{\omega}_b$ and $\dot{\omega}_a$ represent the maximum deceleration.

(3) Constraints on Obstacle Safety Distance

To guarantee the robot's safety throughout its movement, specific constraints are required to maintain a certain distance between the robot and obstacles. In cases where obstacles appear on the predicted trajectory, the robot can reduce its velocity to 0 m/s before encountering the obstacle. The precise constraint formula is provided as Equation (22).

$$v_a = \left\{ (v, \omega) \mid v \leq \sqrt{2dist(v, \omega)v}, \omega \leq \sqrt{2dist(v, \omega)\omega} \right\} \tag{22}$$

where A represents the closest distance between the simulated trajectory corresponding to velocity space B and obstacles.

3.1.3. Adaptive Velocity Adjustment Strategy

In the DWA, the magnitudes of the robot's linear and angular velocities are adjusted based on the difference between the optimal trajectory φ_{best} and the robot's orientation φ_t, ensuring relatively stable speed. In environments with denser obstacles, failing to promptly adjust the speed could lead to excessively small turning angles and increase the risk of collision with obstacles. Therefore, in more intricate settings, a lower linear velocity and a higher angular velocity are employed to enable smoother motion, ensuring that the planned path closely aligns with the optimal path, facilitating quicker attainment of the target point, and enhancing algorithm efficiency. Linear-velocity adjustment and angular-velocity adjustment are represented as follows:

$$\nu_r(t+1) = \nu_r(t) + k_v \frac{\varphi_{\text{path}}(t+1) - \varphi_{\text{path}}(t)}{\varphi_{\max}} \nu_{\max} \tag{23}$$

$$\omega_r(t+1) = \omega_{\max} - k_\omega \frac{\nu_r(t)}{\nu_{\max}} \omega_r(t) \tag{24}$$

where ν_r represents the robot's linear velocity. φ_{path} denotes the robot's navigable area. $\varphi_{\max}$ corresponds to the robot's maximum exploration area. ω_r signifies the robot's angular velocity. k_v and k_ω are dynamic adjustment parameters.

3.1.4. Optimization of DWA's Evaluation Function

To tackle the problem of traditional DWA easily getting stuck in local optima, the global path generated by MISSA is introduced to guide the DWA in local dynamic planning. To make the local dynamic path closer to the global path and more efficiently reach the local sub-goal point, a global path deviation cost evaluation function is added to the evaluation function. This evaluation function combines global path information, calculating the distance between the local motion trajectory and the global path, as illustrated in Figure 3. The formula is as follows:

$$Pathdist(v,\omega) = \min\left(\sum_{i=1}^{N} \sqrt{\left(x_g^i - x_l^i\right)^2 + \left(y_g^i - y_l^i\right)^2}/N\right) \tag{25}$$

where x_g^i and y_g^i are the coordinates of sampling points on the global planning path, and x_l^i and y_l^i are the coordinates of sampling points on the local path trajectory. N represents the total number of sampling points.

In response to dynamic obstacles, an optimization is applied to the distance evaluation function in the traditional DWA. The distance evaluation function is divided into two components, $Dist\_S(v,\omega)$ and $Dist\_D(v,\omega)$, specifically designed to assess the distance between the robot and static obstacles and dynamic obstacles. Different safety distances are established for static and dynamic obstacles.

Consider the coordinates of a static obstacle as (x_s, y_s), and let D_s represent the minimum distance between the motion trajectory at time t and the static obstacle. R denotes the radius of the mobile robot's chassis. For irregularly shaped robots, the maximum chassis radius is taken as R, with an additional 0.15 R added to ensure a safe distance from static obstacles. The constraint formula $Dist\_S(v,\omega)$ is expressed as follows:

$$D_s = \min\left((x_t - x_s)^2 + (y_t - y_s)^2\right) \tag{26}$$

$$Dist\_S(v,\omega) = \begin{cases} \frac{1}{D_s}, D_s \geqslant 1.15R \\ 0, D_s < 1.15R \end{cases} \tag{27}$$

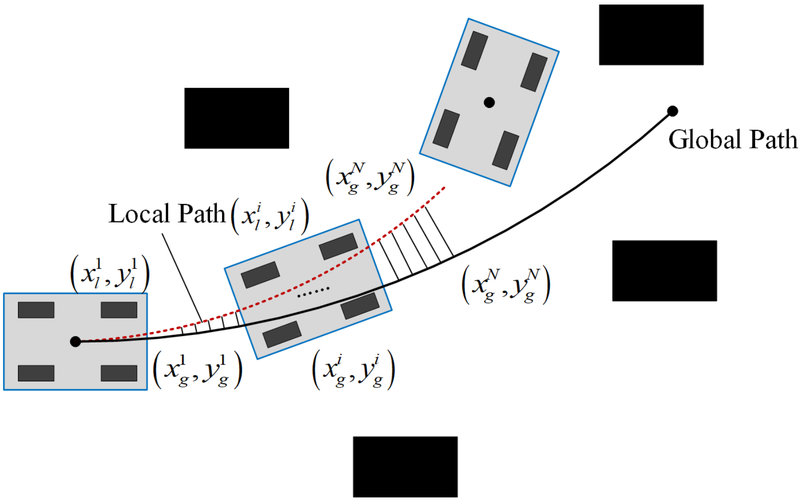


Figure 3. Schematic diagram of the global path deviation cost evaluation function.

Suppose the dynamic obstacle's coordinates are denoted as (x_i, y_i), and let D_d represent the minimum distance between the robot's motion trajectory at time t and the dynamic obstacle. V_t represents the robot's velocity at the time t, while S stands for the robot's path of movement within the time interval Δt. Considering the presence of variable acceleration, this path is expanded to 1.15 times its original length to guarantee a safe distance between the robot and the dynamic obstacle. The constraint formula $Dist\_D(v, \omega)$ is outlined as follows:

$$D_d = \min\left((x_t - x_i)^2 + (y_t - y_i)^2\right) \tag{28}$$

$$Dist\_D(v, \omega) = \begin{cases} \frac{1}{D_s}, D_s \geqslant 1.15S \\ 0, D_s < 1.15S \end{cases} \tag{29}$$

$$S = V_t \cdot \Delta t \tag{30}$$

In conclusion, the improved evaluation function for the DWA is

$$G(v, \omega) = \begin{cases} \chi(\alpha Head(v, \omega) + \beta Vel(v, \omega) + Dist(v, \omega)) \\ Dist(v, \omega) = \gamma Dist\_S(v, \omega) + \sigma Dist\_D(v, \omega) + \varphi Pathdist(v, \omega) \end{cases} \tag{31}$$

where χ represents the smoothing coefficient, while α, β, γ, σ, and φ represent the weighted coefficients of the corresponding functions.

3.2. Fusion Algorithm for Dynamic Path Planning

The optimized MISSA can rapidly acquire global path information, excelling in simple environments with only static obstacles. However, the generated global path lacks smoothness and does not account for newly introduced obstacles, rendering it ineffective against real-time obstacle avoidance in intricate environments. Conversely, the DWA utilizes the robot's sensing range as a window, continuously refreshing it over time to detect the presence of dynamic obstacles in the robot's vicinity, thus possessing a degree of dynamic obstacle avoidance capability. However, the DWA lacks global path guidance and is prone to local optima, leading to failure in reaching the target point. To address these issues, this study combines the MISSA and DWA to propose a new approach to mobile robot dynamic path planning, termed MISSA-DWA. The primary procedure of this algorithm is as follows: In the context of a known global map, MISSA conducts global static path planning. Key points are then extracted from the global path generated by MISSA and are used as local sub-target points for the DWA. The DWA is employed for local dynamic path planning in segments, ensuring the robot's real-time obstacle avoidance. This guarantees that the ultimately generated path is globally optimal and free from collisions. The integrated algorithm effectively leverages the strengths of both methods, reducing the number of path turning points while enhancing path smoothness and safety. The path-planning process of the MISSA-DWA fusion algorithm is depicted in Figure 4.

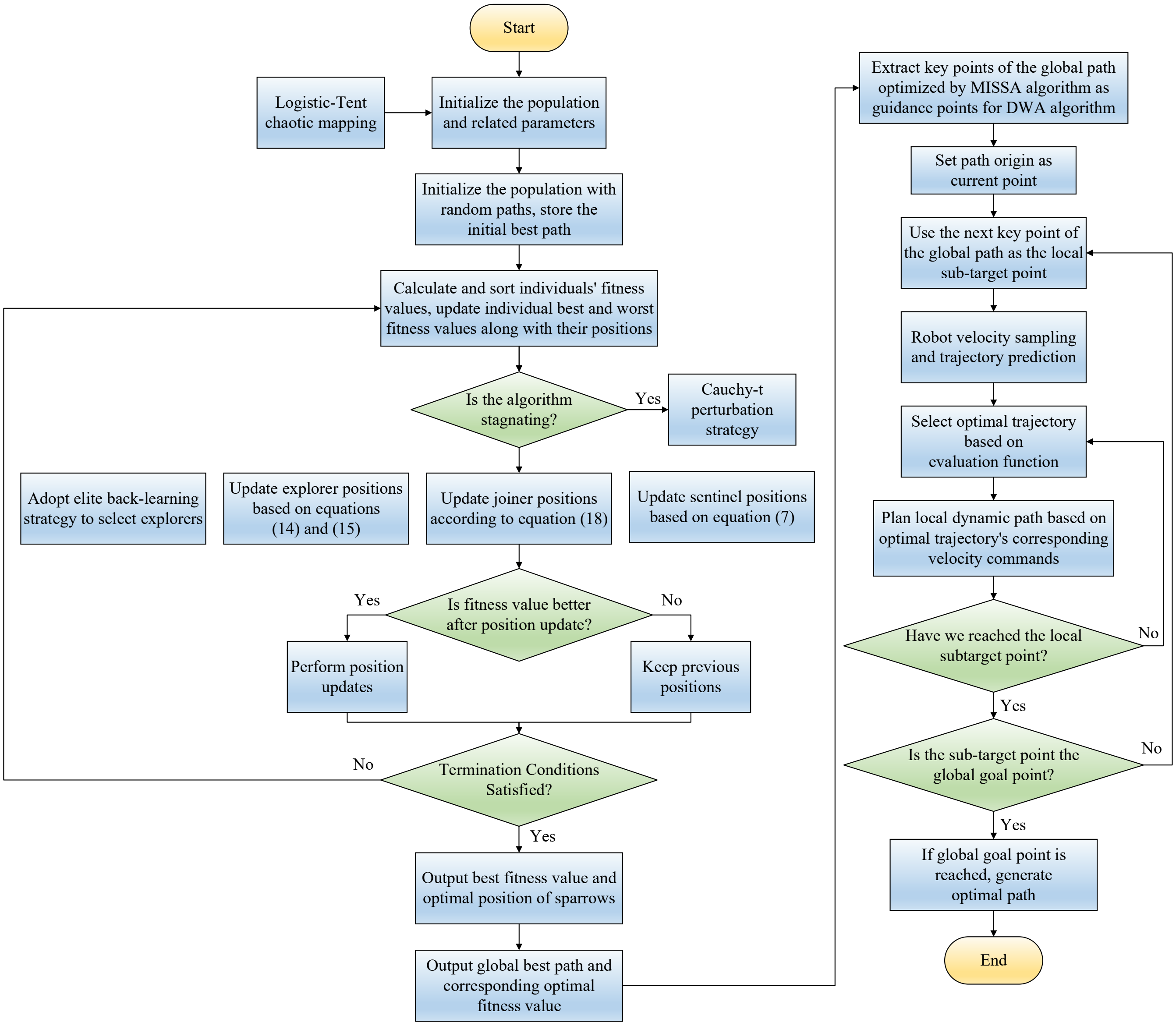

Start
Logistic-Tent chaotic mapping
Initialize the population and related parameters
Initialize the population with random paths, store the initial best path
Calculate and sort individuals' fitness values, update individual best and worst fitness values along with their positions
Is the algorithm stagnating?
Yes
Cauchy-t perturbation strategy
Adopt elite back-learning strategy to select explorers
Update explorer positions based on equations (14) and (15)
Update joiner positions according to equation (18)
Update sentinel positions based on equation (7)
Yes
Is fitness value better after position update?
No
Perform position updates
Keep previous positions
No
Termination Conditions Satisfied?
Yes
Output best fitness value and optimal position of sparrows
Output global best path and corresponding optimal fitness value
Extract key points of the global path optimized by MISSA algorithm as guidance points for DWA algorithm
Set path origin as current point
Use the next key point of the global path as the local sub-target point
Robot velocity sampling and trajectory prediction
Select optimal trajectory based on evaluation function
Plan local dynamic path based on optimal trajectory's corresponding velocity commands
Have we reached the local subtarget point?
No
Yes
Is the sub-target point the global goal point?
No
Yes
If global goal point is reached, generate optimal path
End

Figure 4. Path-planning process diagram of the MISSA-DWA fusion algorithm.

4. Simulation and Experimental Results Analysis

4.1. Environment Modeling

Environment modeling is a crucial step in robot path planning. Common methods for environmental modeling include grid-based methods, topological methods, and visibility graph methods [44]. This study adopts a grid-based approach for environmental representation. In the grid map, grid cells are binarized according to Equation (32) to establish the grid map. A value of 0 represents free traversable space, while 1 denotes obstacles. An illustrative grid map example is depicted in Figure 5, where the workspace is divided into white and black grid cells. Here, black grid cells symbolize obstacle regions that are inaccessible to the robot, while white grid cells represent areas traversable by the robot. Therefore, the task of charting paths for mobile robots within complex environments transforms into the objective of determining the most efficient route from the starting point to the destination, utilizing a grid map model under defined constraints [45,46].

$$map(i,j)\begin{cases} 0, \text{Free space} \\ 1, \text{Obstacle} \end{cases} \tag{32}$$

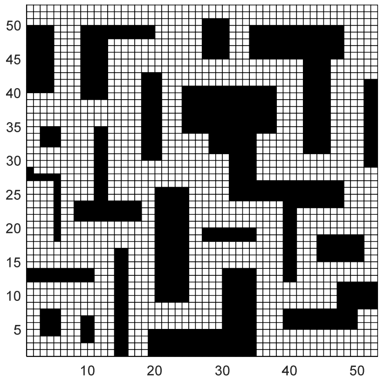


Figure 5. Grid-based terrain model.

4.2. Simulation Experimental Environment

To guarantee the integrity and comparability of our experimental results, all simulation experiments adhered to uniform configurations. A grid-based approach was employed to construct grid maps for the simulation experiments. The computer processor used was an AMD R7-6800H with a clock frequency of 3.20 GHz, the operating system was Windows 11 (64-bit), and the system had 16 GB of memory. MATLAB R2022b served as the simulation platform.

4.3. Simulation Comparative Experiment of MISSA Global Path Planning

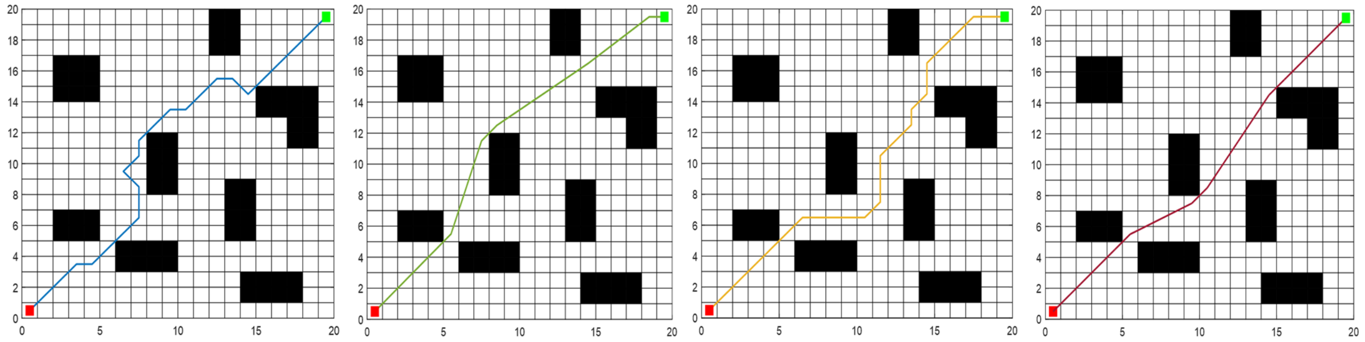
To validate the superior performance of the MISSA, this paper juxtaposes it with the conventional ACO, improved ant colony optimization (IACO), and SSA by performing simulations on two grid maps with distinct complexities. Simulation analysis is conducted on 20 × 20 grid maps of varying complexities. Crucially, to authenticate the algorithm's efficiency, parameter configurations for both the SSA and the MISSA are kept uniform. Table 1 delineates the parameter specifications for both the sparrow search and ant colony optimization algorithms.

Table 1. Algorithm parameter settings.

SSA	Number of Sparrows	Maximum Iterations	Search Dimensions	Alarm Threshold	Producer Ratio	Producer Ratio
Parameters	60	300	16	0.8	0.3	0.2
ACO	Number of Ants	Maximum Iterations	Pheromone Factor	Heuristic Function Factor	Pheromone Evaporation Factor	Pheromone Constant
Parameters	60	300	2	6	0.1	5

4.3.1. Environment Model 1

In environment model 1, obstacles were sparsely distributed and simulations were executed on a 20 × 20 grid map within MATLAB. Figure 6 visually represents the path-planning outcomes of ACO, IACO, SSA, and MISSA. To minimize the impact of random variables, each algorithm underwent 50 repetitive simulation trials. The comparative performance evaluation of these four algorithms is encapsulated in Table 2.



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

Figure 6. The trajectories of four path-planning algorithms within environment model 1: (**a**) ACO; (**b**) IACO; (**c**) SSA; (**d**) MISSA. The initial point is depicted by a red dot, while the termination point is denoted by a green dot.

Table 2. Comparison of performance indicators for algorithms (ACO, IACO, SSA, and MISSA) under environment model 1.

Algorithm	**Path Length/m**		**Total Rotation Angle**		**Time/s**	
	Minimum	**Average**	**Minimum**	**Average**	**Minimum**	**Average**
ACO	28.1416	30.3926	425	743.45	0.4178	1.8917
IACO	27.9073	29.8455	126	187.37	2.9614	8.5572
SSA	28.3411	32.7429	218	318.74	0.3796	1.7083
ISSA	27.1290	27.7306	48	61.85	0.3282	0.9935

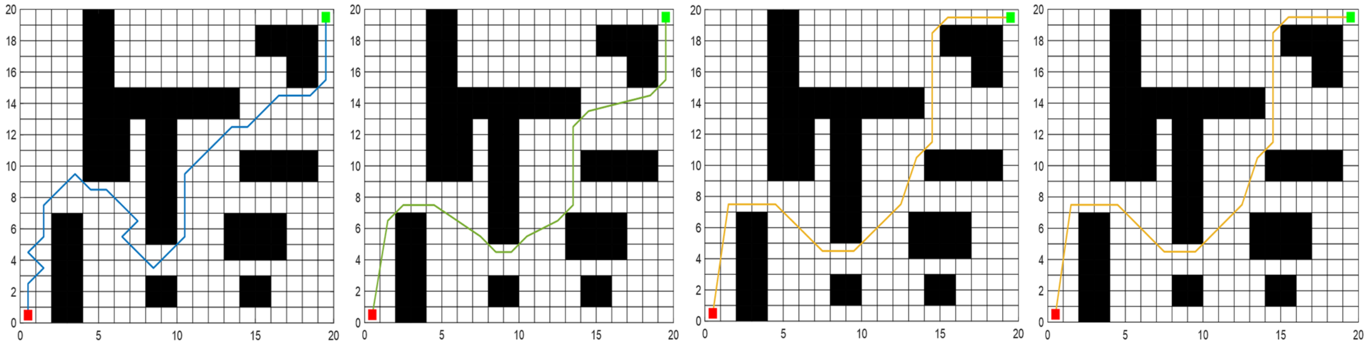
Simulation outcomes reveal that all four algorithms are adept at devising collision-free global paths. Analyzing path length, total rotation angle, and runtime from 50 simulation trials, the MISSA emerged superior in path efficiency, charting the shortest path of 27.1290 m. This trims the shortest path lengths determined by the ACO, IACO, and SSA by 3.60%, 2.79%, and 4.28%, respectively. Its average path lengths also undercut theirs by 8.76%, 7.09%, and 15.31%, respectively. In rotation metrics, MISSA exhibited commendable precision. It slashed the minimal total rotation angle relative to paths from ACO, IACO, and SSA by 88.71%, 61.90%, and 77.98%, respectively. Similarly, the average rotation angles were curtailed by 91.68%, 66.99%, and 80.60%. When clocking runtime, MISSA proved to be more time-efficient. Its minimal runtime surpassed ACO, IACO, and SSA by shrinking durations by 21.45%, 88.92%, and 13.54%, respectively. Likewise, average runtimes dwindled by 47.48%, 88.39%, and 41.84% compared to the other algorithms.

4.3.2. Environment Model 2

To bolster the assertion of the enhanced algorithm's supremacy, more complex obstacles are added to the grid map for simulation experiments. Figure 7 distinctly displays the path optimization outcomes for ACO, IACO, SSA, and MISSA. After conducting 50 iterative experiments, Table 3 provides a comprehensive performance comparison of these four algorithms.

From the simulation results of the four algorithms, the MISSA's planned path stands out as the shortest, measuring at 34.3719 m. This is a reduction of 9.24%, 6.15%, and 4.98% compared to the shortest paths from the ACO, IACO, and SSA, respectively. In terms of average path lengths, the MISSA outperforms with a reduction of 19.55%, 9.81%, and 11.92% relative to the aforementioned algorithms. Regarding rotation angles, the MISSA's minimal total rotation angle is 57.75%, 35.96%, and 20.86% less than the ACO, IACO, and SSA paths, respectively. The average rotation angles also show marked improvement, declining by 77.19%, 69.17%, and 57.79% for the respective algorithms. In runtime efficiency, the MISSA demonstrates significant prowess. Its fastest runtime is shorter by 69.77%, 82.71%, and

20.70% compared to the ACO, IACO, and SSA. The average runtimes also see notable reductions of 71.03%, 86.53%, and 33.23% compared to the trio of algorithms.



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

Figure 7. The trajectories of four path-planning algorithms within environment model 2: (**a**) ACO; (**b**) IACO; (**c**) SSA; (**d**) MISSA. The initial point is depicted by a red dot, while the termination point is denoted by a green dot.

Table 3. Comparison of performance indicators for algorithms (ACO, IACO, SSA, and MISSA) under environment model 2.

Algorithm	**Path Length/m**		**Total Rotation Angle**		**Time/s**	
	Minimum	**Average**	**Minimum**	**Average**	**Minimum**	**Average**
ACO	37.8701	43.3414	748	1436.17	1.8795	4.5309
IACO	36.6226	38.6631	585	1062.82	3.2852	9.7443
SSA	36.1751	39.5893	472	776.16	0.7164	1.9656
ISSA	34.3719	34.8701	316	327.63	0.5681	1.3124

From the aforementioned experimental results, it is evident that the MISSA introduced in this study is adept at crafting globally optimal paths across diverse grid map complexities. The generated global path consistently outperforms the other three benchmarked algorithms, excelling in metrics like path length, total rotation angle, and runtime. Beyond its superior path optimization prowess, the MISSA also showcases remarkable stability and the ability to produce smooth paths.

4.4. Analysis of Random Obstacle Avoidance in the Fusion Algorithm

In a grid map of the same environment, four sets of simulation experiments were designed to validate the integrated algorithm. The path planning was conducted using both the MISSA and the combined MISSA-DWA method. The starting and ending grids were marked as "S" and "G", respectively. Various quantities of random obstacles, represented as blue rectangles, were added to the grid map. The parameters for the DWA are set as follows: the maximum linear velocity is 1 m/s, the maximum linear acceleration is 0.3 $\mathrm{m/s^2}$, the maximum angular velocity is 30 °/s, the maximum angular acceleration is 50 $^{\circ}/\mathrm{s}^2$, the velocity resolution is 0.01 m/s, the angular velocity resolution is 1.0 °/s, the time resolution is 0.1 s, and the prediction period is 3.0 s. The parameters for the evaluation function are set as $\alpha = 0.15$, $\beta = 0.2$, $\gamma = 0.15$, $\sigma = 0.25$, and $\varphi = 0.3$.

Figure 8 displays the simulation outcomes for random obstacle avoidance. The path plotted by the MISSA is illustrated in red, whereas the blue pathway symbolizes the path designed using the MISSA-DWA fusion method. From these simulations, it becomes evident that in the absence of random obstacles, both algorithms can achieve optimal paths. However, when confronted with random obstacles, the MISSA, relying solely on global map data, fails to recognize and subsequently avoid these obstructions. Conversely, the MISSA-DWA approach seamlessly integrates real-time obstacle evasion for sporadic barriers while ensuring path fluidity. As the quantity of random obstacles rises, the designed route consistently demonstrates its efficacy in bypassing impediments and securing a successful arrival at the intended destination. In essence, the introduced hybrid methodology proficiently negotiates random obstacles within grid maps, delivering a streamlined route that closely mirrors the globally optimal trajectory. Pertinent performance metrics for random obstacle sidestepping, utilizing the fusion algorithm, are detailed in Table 4.

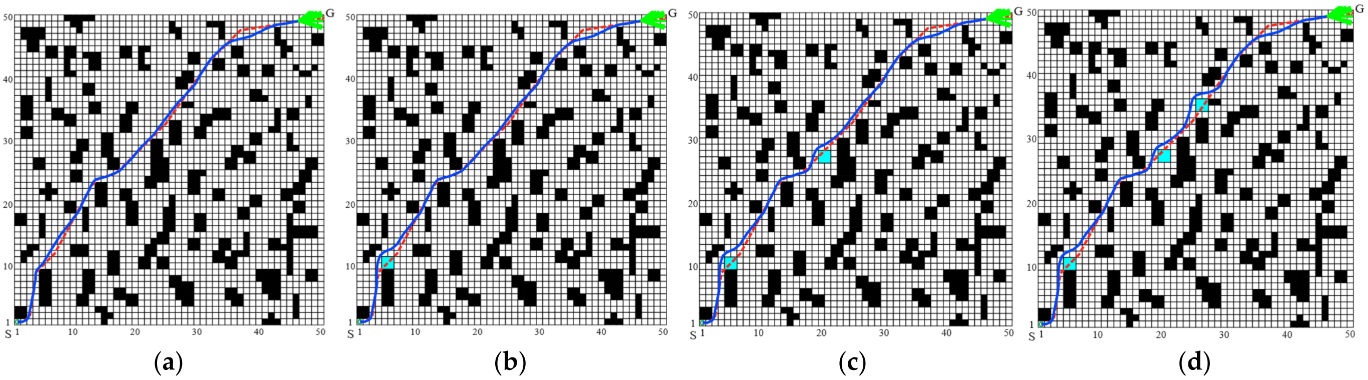


Figure 8. Diagram illustrating the trajectory for random obstacle avoidance in path-planning scenarios: (**a**) without random obstacles; (**b**) with one added random obstacle; (**c**) including two random obstacles; (**d**) including two random obstacles.

Table 4. Path-planning performance of the MISSA-DWA fusion algorithm.

Quantity of Random Obstacles	Path Length/m	Time/s
No random obstacles	74.155	106.58
One random obstacle	74.763	109.69
Two random obstacles	75.232	113.73
Three random obstacles	76.206	117.62

4.5. Dynamic Path Planning with the Fusion Algorithm

In an effort to robustly assess the obstacle evasion capabilities of the MISSA-DWA fusion algorithm within dynamic environments, simulations were executed across four types of environmental models on a 50 × 50 grid map. A certain number of dynamic obstacles and random unknown obstacles are introduced into the experimental environment. "S" and "G" identify the starting and ending grid cells, respectively. The green cursor reflects the robot's directional sensing. Dynamic obstacles are illustrated as pink orbs, with their origins and movement paths depicted by light pink dots and dashed lines. Blue rectangles represent unpredictable obstructions. The trajectory crafted by the MISSA is denoted in red, while the fusion algorithm's real-time navigation is rendered in blue, emphasizing the fusion algorithm's agility in swiftly adapting to dynamic changes.

From Figure 9a, it is evident that the robot follows the path prescribed by the MISSA until it encounters the first unknown obstacle. Upon detecting this unknown static obstacle and assessing the robot's relative position and motion status in relation to it, there is a foreseen risk of a collision along the original trajectory. Consequently, the fusion algorithm is activated to engage in local obstacle avoidance, effectively circumventing the unknown obstacle. Subsequently, the robot recalibrates its route to converge with the optimal path generated by the MISSA. As illustrated in Figure 9b, the robot detects a dynamic obstacle at that location and, foreseeing a potential side collision, employs a local obstacle avoidance strategy after determining the dynamic obstacle's movement direction. Through four instances of real-time obstacle avoidance scenarios involving both unknown static and dynamic obstacles, the robot successfully reaches the target point, as evidenced in Figure 9c.

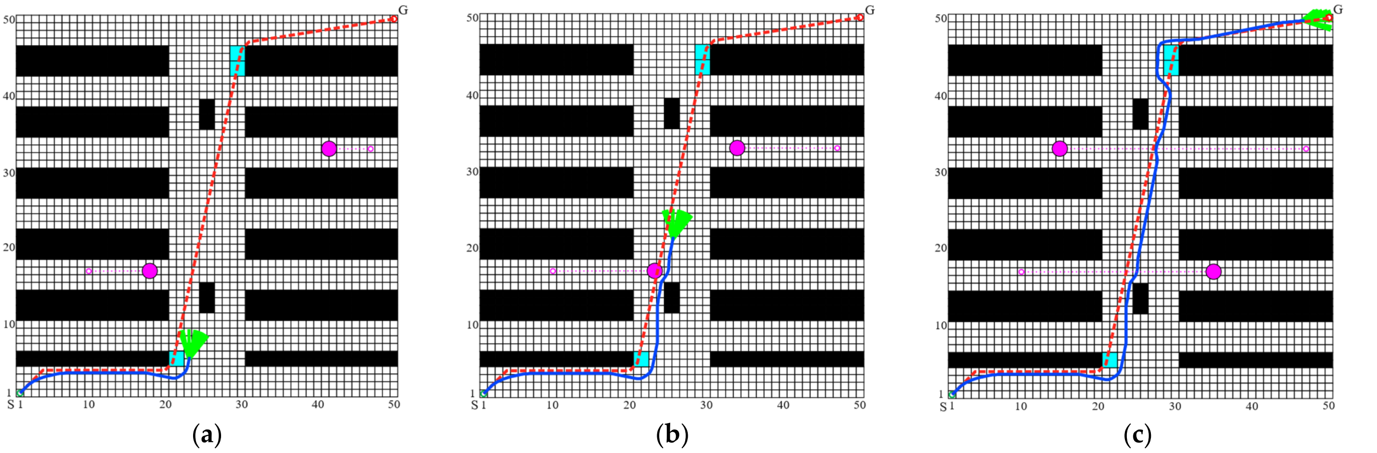


Figure 9. Dynamic obstacle avoidance effect of the MISSA-DWA fusion algorithm in environment 1: (**a**) avoiding the random obstacle; (**b**) avoiding the dynamic obstacle; (**c**) completed dynamic path planning to reach the target point.

In Figure 9, it is evident that the robot, commencing its journey from the initial point, consistently demonstrates the ability to promptly and accurately employ appropriate obstacle avoidance strategies when confronted with lateral and frontal collisions involving random unknown obstacles and dynamic obstacles along the global path. This capability enables the robot to discern a smooth and secure path, ultimately achieving the anticipated outcome of algorithm fusion.

To further substantiate the efficacy of the proposed fusion algorithm, a series of dynamic path-planning simulations were executed under diverse scenarios. The paths generated by the fusion algorithm were juxtaposed with the globally planned paths determined by the MISSA to ascertain the dynamic obstacle avoidance performance outlined in this paper. The simulation results encompassing three distinct complexity levels of environments are illustrated in Figure 10.

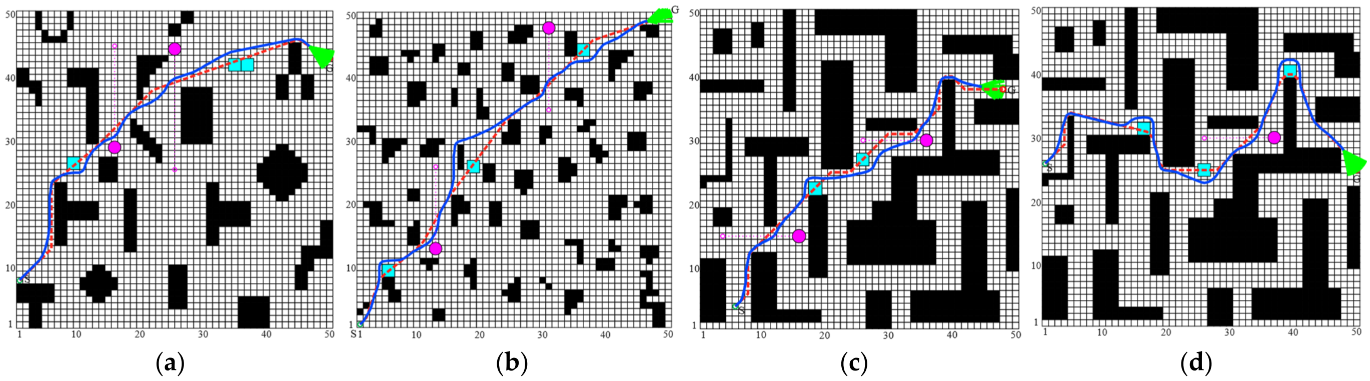


Figure 10. MISSA-DWA fusion algorithm's dynamic obstacle avoidance effect. (**a**) Simulation diagram 1 in environment 2; (**b**) simulation diagram 2 in environment 3; (**c**) simulation diagram 3 in environment 4; (**d**) simulation diagram 4 in environment 4.

Summarizing the simulation results across the four environments, it can be inferred that in environments containing both unknown static and dynamic obstacles, the enhanced MISSA's planning process, which does not rely on dynamic grid information, generates paths consistent with those in static environments. As a consequence, it lacks real-time dynamic obstacle avoidance capabilities, rendering it incapable of maneuvering around dynamic obstacles. In contrast, the proposed MISSA-DWA algorithm, coupled with the improved DWA algorithm, demonstrates the ability to perform real-time local dynamic path planning based on the global path. It successfully navigates around newly introduced unknown static and dynamic obstacles, crafting smooth paths that strike a balance between global optimality and effective obstacle avoidance. The algorithm adapts seamlessly to complex dynamic obstacle environments, thereby enhancing the operational efficiency of the robot.

4.6. Experimental Validation

To further assess the effectiveness of the algorithm in real-world applications, practical robot path-planning experiments were conducted in an actual environment. The current study utilized the Songling Robot SCOUT2.0 robot chassis, equipped with a Velodyne 16-line LiDAR sensor and integrated with the ROS system on the host computer. The host computer facilitated control over the robot and its sensors, enabling the collection of environmental data. The Gmapping-SLAM algorithm was employed to generate a two-dimensional grid map of the experimental surroundings. The resulting grid map was saved in image format, with white areas denoting explored and unoccupied regions, black areas indicating detected obstacles, and gray areas representing inflated obstacle boundaries. Subsequently, the MISSA and MISSA-DWA fusion algorithms were validated within the constructed grid map.

The experiments in a real-world environment are divided into static and dynamic components. In the static path-planning experiment, the first step involves constructing the environmental map and launching the Rviz platform. Subsequently, the robot's initial and destination points are specified, and the MISSA-DWA fusion algorithm is invoked to facilitate global path planning for the robot, guiding it from the starting point to the target point within a static environment. The robot adheres to the planned path as it progresses toward the destination. The experimental procedure, as depicted in Figures 11 and 12, illustrates that the mobile robot departs from its initial position and adeptly navigates around all static obstacles throughout the entire movement process. Eventually, the robot safely arrives at the target point by following the globally optimal path. The feedback of robot control parameters during the experiment is shown in Figure 13.

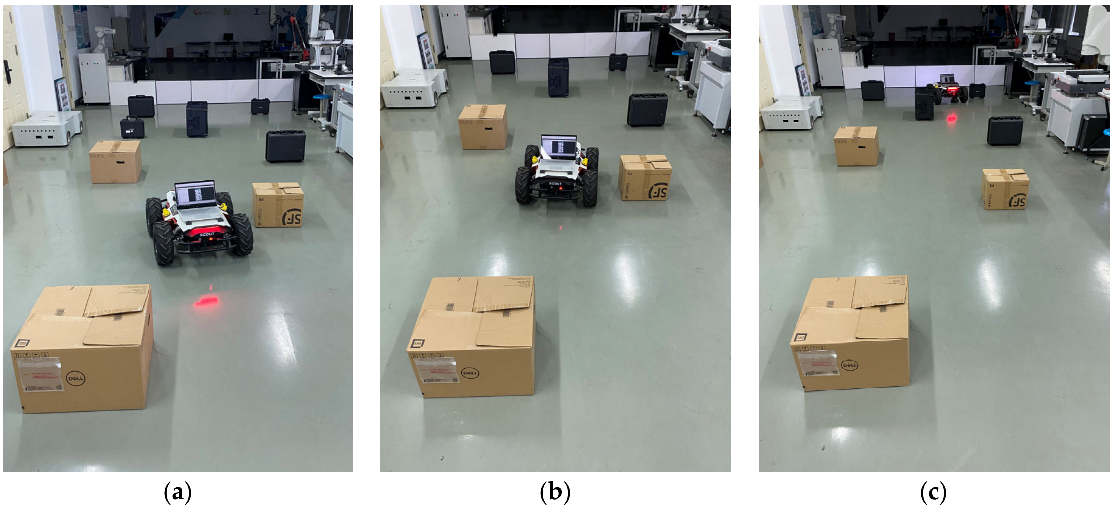


Figure 11. Obstacle avoidance process in a static environment.(**a**) obstacle avoidance process 1; (**b**) obstacle avoidance process 2; (**c**) reaching the target point.

To assess the robot's capability to avoid dynamic obstacles during its movement, a small mobile robot was introduced as a dynamic obstacle in the experimental environment, as depicted in Figure 14. The experimental procedure, as illustrated in Figures 14 and 15, demonstrates that after the robot plans its global path and embarks on its journey, the dynamic obstacle robot is activated and positioned onto the pre-planned global path. As the dynamic obstacle enters the detection range of the local path-planning algorithm, the robot promptly identifies and responds, utilizing the fusion algorithm for local dynamic obstacle avoidance. This enables successful avoidance of the dynamic obstacle, as depicted in Figure 15a,c. Subsequently, the robot continues its movement toward the planned global optimal path and ultimately reaches the target point. The experimental results in Figures 14 and 15 illustrate that the proposed fusion algorithm ensures smooth motion paths while effectively evading dynamic obstacles in the environment, thereby validating the algorithm's effectiveness. The feedback of robot control parameters during the experiment is shown in Figure 16.

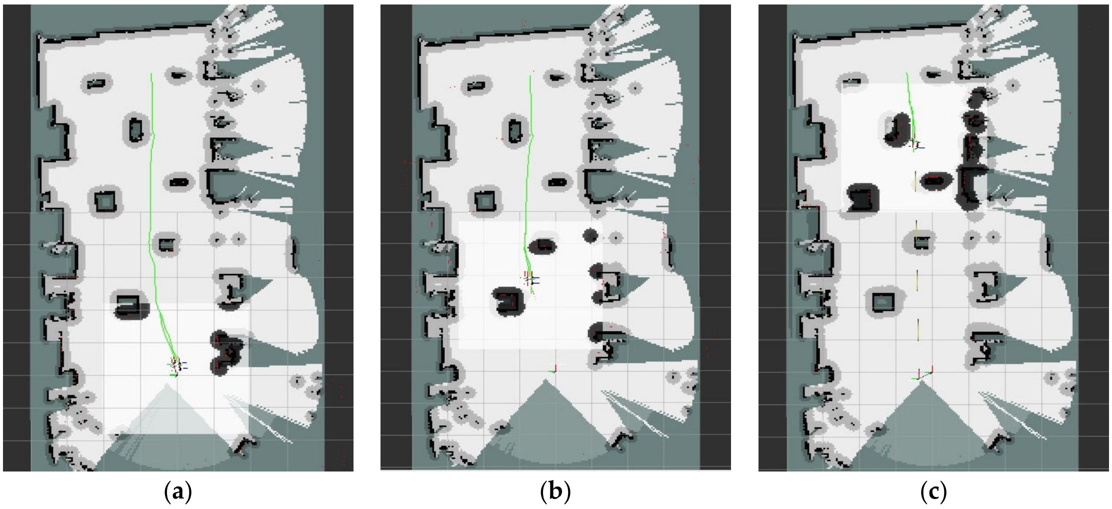


Figure 12. Path planning in a static environment. (**a**) the mobile robot starts to move (**b**) obstacle avoidance process 1. (**c**) obstacle avoidance process 2.

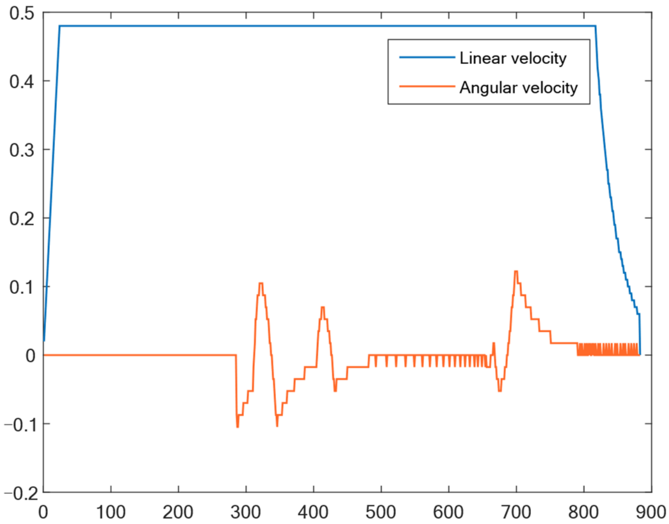


Figure 13. Output results of linear velocity and angular velocity in a static environment.

To further validate the proposed algorithm and its effectiveness, experiments were conducted in a more complex map environment. Additional randomly shaped static obstacles and dynamic obstacles were introduced into the experimental environment. In the more complex conditions, as shown in Figures 17 and 18, two blue benches and a barrel were added as new random static obstacles, and a small mobile robot served as a dynamic obstacle. The experimental results demonstrate that the mobile robot can still plan its path reasonably based on the algorithm and navigate around static and dynamic obstacles effectively, ensuring the robot reaches the target point safely. This further validates the effectiveness of the algorithm. The feedback of robot control parameters during the experiment is shown in Figure 19.

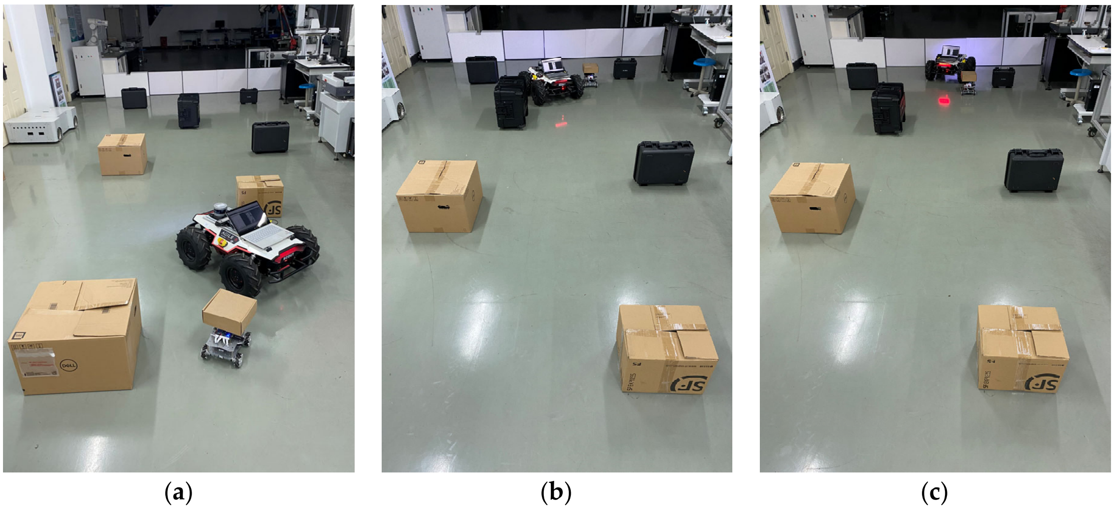


Figure 14. Obstacle avoidance process in a dynamic environment 1. (**a**) obstacle avoidance process 1. (**b**) obstacle avoidance process 2. (**c**) reaching the target point.

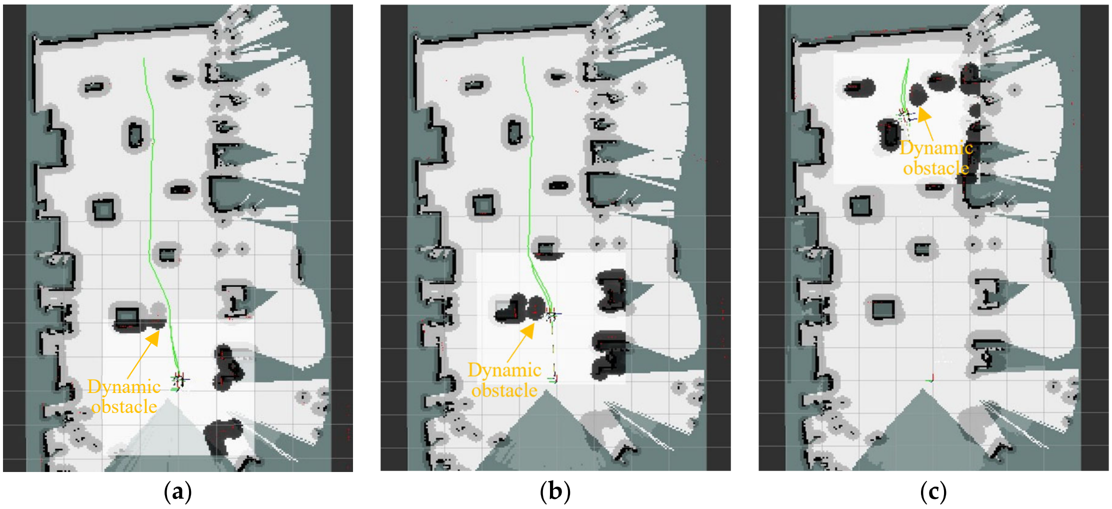


Figure 15. Path planning in dynamic environment 1. (**a**) the mobile robot starts to move (**b**) obstacle avoidance process 1. (**c**) obstacle avoidance process 2.

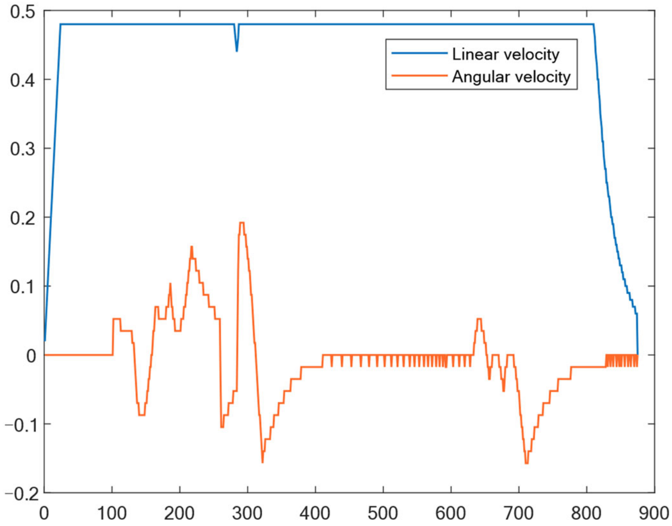


Figure 16. Output results of linear-velocity and angular-velocity dynamic environment 1.

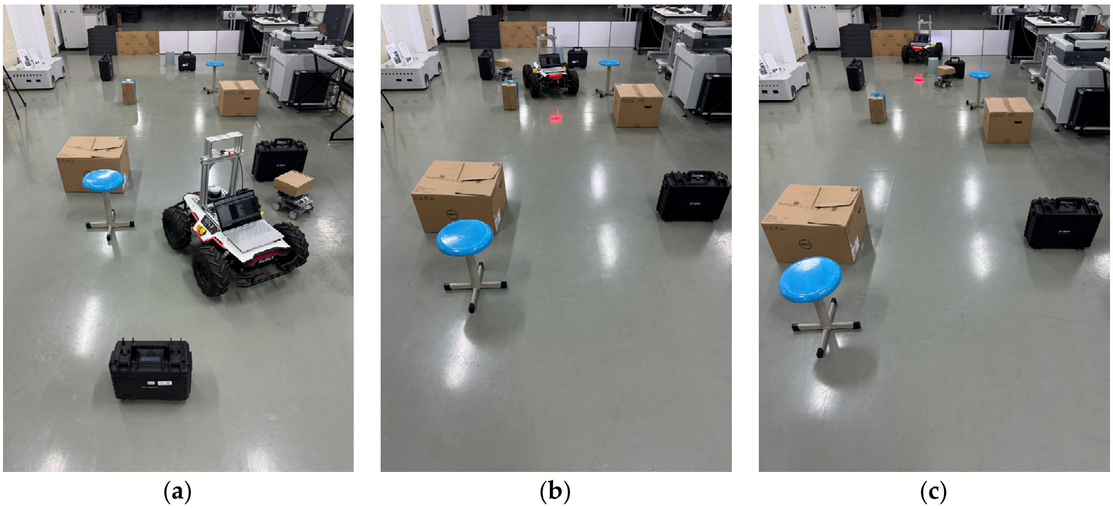


Figure 17. Obstacle avoidance process in dynamic environment 2. (**a**) obstacle avoidance process 1. (**b**) obstacle avoidance process 2. (**c**) reaching the target point.

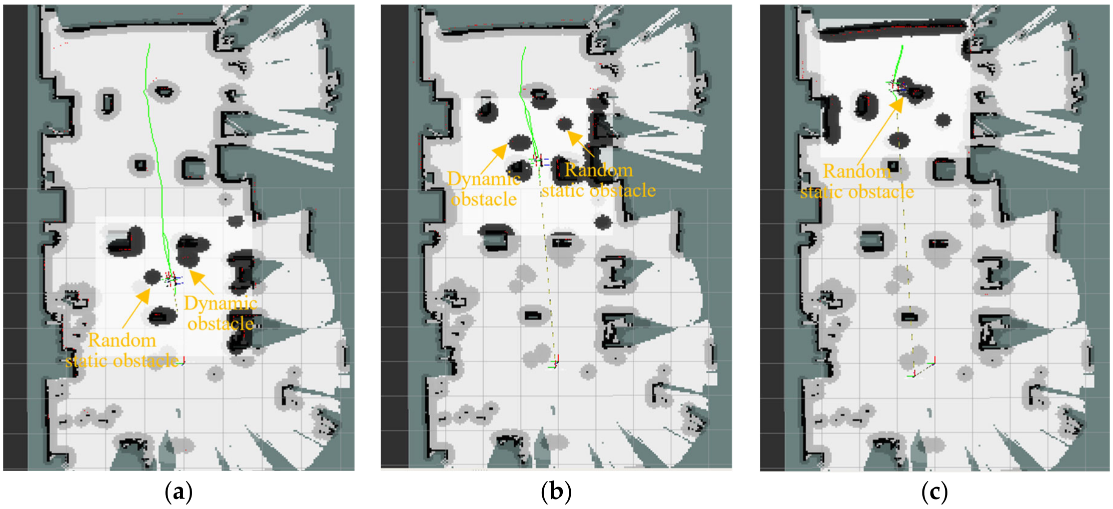


Figure 18. Path planning in a dynamic environment 2. (**a**) obstacle avoidance process 1. (**b**) obstacle avoidance process 2. (**c**) obstacle avoidance process 3.

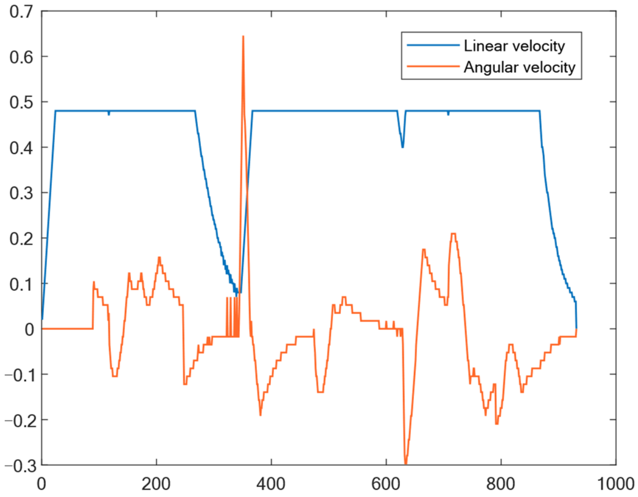


Figure 19. Output results of linear-velocity and angular-velocity dynamic environment 2.

5. Conclusions

In response to the shortcomings and limitations of the basic SSA in path planning, as well as the challenges posed by dynamic path planning for mobile robots, this study introduces a novel approach that combines the enhanced SSA with the DWA. Through a comparative analysis of path length, total rotation angle, and algorithm execution time for static path planning, along with the execution of obstacle avoidance experiments in dynamic path planning under various scenarios, the following conclusions can be drawn:

- Initially, logistic–tent chaotic mapping is utilized to initialize the sparrow population, generating a relatively evenly distributed and stable initial population. Subsequently, an elite reverse learning strategy is employed to select producers, which forms the foundation for algorithm stability. Following this, a dynamic self-adaptive position update strategy is integrated to enhance the optimization capability of the algorithm. This improvement involves refining the position update formulas for both producers and scroungers. Additionally, the Lévy flight strategy is applied to update the positions of scroungers, accompanied by the incorporation of a best position perturbation strategy to further enhance the algorithm's ability to escape local optima. Through comparative experiments in static environments and obstacle avoidance experiments in various dynamic environment scenarios, the results demonstrate that the global paths planned by MISSA outperform those generated by the other three comparative algorithms in terms of path length, total rotation angle, and algorithm runtime. These findings underscore MISSA's robust path optimization capability and stability.
- The evaluation function of the DWA is refined, and an adaptive velocity adjustment strategy is introduced. Subsequently, MISSA is integrated with the optimized DWA, utilizing the key points of the global path generated by MISSA as local sub-goals for the DWA. This integration results in the development of MISSA-DWA. Through path-planning experiments conducted in diverse dynamic environments and scenarios, it is concluded that the integrated MISSA-DWA algorithm not only devises globally optimal paths but also excels in generating smooth paths while dynamically navigating around obstacles. It effectively avoids unidentified obstructions, ensuring the secure arrival of the robot at the target destination. This enhancement significantly improves the efficiency and safety of the robot's movement.

Moreover, the research in this paper still has certain limitations. The main directions for future research plans are twofold. In terms of algorithms, researchers should consider trying other strategies to further optimize the algorithm's performance. Simultaneously, researchers should comprehensively consider factors such as the traversal cost of different areas, changes in terrain height, and the different costs of uphill and downhill traversal in the environmental map to enhance the practicality of the algorithm. On the application front, there is a contemplation of applying the algorithm to multi-objective planning and multi-robot planning issues.

Author Contributions: Conceptualization, J.H., W.J. and L.Y.; methodology, J.H., W.J., Z.L. and L.Y.; software, J.H.; validation, J.H. and W.J.; investigation, J.H.; resources, Z.L.; data curation, J.H., W.J. and L.Y.; writing—original draft preparation, J.H. and W.J.; writing—review and editing, J.H. and W.J.; project administration, W.J., Z.L., X.H. and B.G.; supervision and funding acquisition, Z.L., X.H. and B.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (Grant No. 52005471 and 52075511) and the National Key Research and Development Program of China (Grant No. 2022YFF0705704).

Data Availability Statement: The data used to support the findings of this study are available from the first author upon request.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Duan, L.-M. Path Planning for Batch Picking of Warehousing and Logistics Robots Based on Modified A* Algorithm. *Int. J. Onl. Eng.* **2018**, *14*, 176. [CrossRef]
2. Tong, X.; Yu, S.; Liu, G.; Niu, X.; Xia, C.; Chen, J.; Yang, Z.; Sun, Y. A Hybrid Formation Path Planning Based on A* and Multi-Target Improved Artificial Potential Field Algorithm in the 2D Random Environments. *Adv. Eng. Inform.* **2022**, *54*, 101755. [CrossRef]
3. Zhang, Z.; Chen, J.; Guo, Q. Application of Automated Guided Vehicles in Smart Automated Warehouse Systems: A Survey. *Comput. Model. Eng. Sci.* **2022**, *134*, 1529–1563. [CrossRef]
4. Suresh, K.S.; Venkatesan, R.; Venugopal, S. Mobile Robot Path Planning Using Multi-Objective Genetic Algorithm in Industrial Automation. *Soft Comput.* **2022**, *26*, 7387–7400. [CrossRef]
5. Bai, X.; Jiang, H.; Cui, J.; Lu, K.; Chen, P.; Zhang, M. UAV Path Planning Based on Improved A∗ and DWA Algorithms. *Int. J. Aerosp. Eng.* **2021**, *2021*, 4511252. [CrossRef]
6. Hu, X.; Luo, Z.; Jiang, W. AGV Localization System Based on Ultra-Wideband and Vision Guidance. *Electronics* **2020**, *9*, 448. [CrossRef]
7. Zhong, X.; Tian, J.; Hu, H.; Peng, X. Hybrid Path Planning Based on Safe A* Algorithm and Adaptive Window Approach for Mobile Robot in Large-Scale Dynamic Environment. *J. Intell. Robot. Syst.* **2020**, *99*, 65–77. [CrossRef]
8. Soltani, A.R.; Tawfik, H.; Goulermas, J.Y.; Fernando, T. Path Planning in Construction Sites: Performance Evaluation of the Dijkstra, A∗, and GA Search Algorithms. *Adv. Eng. Inform.* **2002**, *16*, 291–303. [CrossRef]
9. Han, X.; Zhang, X.; Zhang, H. Trajectory Planning of USV: On-Line Computation of the Double S Trajectory Based on Multi-Scale A* Algorithm with Reeds–Shepp Curves. *J. Mar. Sci. Eng.* **2023**, *11*, 153. [CrossRef]
10. Jhong, B.-G.; Chen, M.-Y. An Enhanced Navigation Algorithm with an Adaptive Controller for Wheeled Mobile Robot Based on Bidirectional RRT. *Actuators* **2022**, *11*, 303. [CrossRef]
11. Wang, Q.; Li, J.; Yang, L.; Yang, Z.; Li, P.; Xia, G. Distributed Multi-Mobile Robot Path Planning and Obstacle Avoidance Based on ACO–DWA in Unknown Complex Terrain. *Electronics* **2022**, *11*, 2144. [CrossRef]
12. Chen, T.J.; Sun, Y.; Dai, W.; Tao, W.; Liu, S. On the Shortest and Conflict-Free Path Planning of Multi-AGV System Based on Dijkstra Algorithm and the Dynamic Time-Window Method. *Adv. Mater. Res.* **2013**, *645*, 267–271. [CrossRef]
13. Li, X.; Jiao, T.; Ma, J.; Duan, D.; Liang, S. LSDA-APF: A Local Obstacle Avoidance Algorithm for Unmanned Surface Vehicles Based on 5G Communication Environment. *Comput. Model. Eng. Sci.* **2023**, *138*, 595–617. [CrossRef]
14. Wang, J.; Luo, Y.; Tan, X. Path Planning for Automatic Guided Vehicles (AGVs) Fusing MH-RRT with Improved TEB. *Actuators* **2021**, *10*, 314. [CrossRef]
15. Miao, C.; Chen, G.; Yan, C.; Wu, Y. Path Planning Optimization of Indoor Mobile Robot Based on Adaptive Ant Colony Algorithm. *Comput. Ind. Eng.* **2021**, *156*, 107230. [CrossRef]
16. Sui, F.; Tang, X.; Dong, Z.; Gan, X.; Luo, P.; Sun, J. ACO+PSO+A*: A Bi-Layer Hybrid Algorithm for Multi-Task Path Planning of an AUV. *Comput. Ind. Eng.* **2023**, *175*, 108905. [CrossRef]
17. Gul, F.; Rahiman, W.; Alhady, S.S.N.; Ali, A.; Mir, I.; Jalil, A. Meta-Heuristic Approach for Solving Multi-Objective Path Planning for Autonomous Guided Robot Using PSO–GWO Optimization Algorithm with Evolutionary Programming. *J. Ambient. Intell. Hum. Comput.* **2021**, *12*, 7873–7890. [CrossRef]
18. Liang, J.-H.; Lee, C.-H. Efficient Collision-Free Path-Planning of Multiple Mobile Robots System Using Efficient Artificial Bee Colony Algorithm. *Adv. Eng. Softw.* **2015**, *79*, 47–56. [CrossRef]
19. Zheng, Y.; Luo, Q.; Wang, H.; Wang, C.; Chen, X. Path Planning of Mobile Robot Based on Adaptive Ant Colony Algorithm. *J. Intell. Fuzzy Syst.* **2020**, *39*, 5329–5338. [CrossRef]
20. Raj, R.; Kos, A. An Optimized Energy and Time Constraints-Based Path Planning for the Navigation of Mobile Robots Using an Intelligent Particle Swarm Optimization Technique. *Appl. Sci.* **2023**, *13*, 9667. [CrossRef]
21. Dong, L.; Yuan, X.; Yan, B.; Song, Y.; Xu, Q.; Yang, X. An Improved Grey Wolf Optimization with Multi-Strategy Ensemble for Robot Path Planning. *Sensors* **2022**, *22*, 6843. [CrossRef]
22. Eshtehardian, S.A.; Khodaygan, S. A Continuous RRT*-Based Path Planning Method for Non-Holonomic Mobile Robots Using B-Spline Curves. *J. Ambient. Intell. Hum. Comput.* **2023**, *14*, 8693–8702. [CrossRef]
23. Zhang, Y.; Wen, Y.; Tu, H. A Method for Ship Route Planning Fusing the Ant Colony Algorithm and the A* Search Algorithm. *IEEE Access* **2023**, *11*, 15109–15118. [CrossRef]
24. Xue, J.; Shen, B. A Novel Swarm Intelligence Optimization Approach: Sparrow Search Algorithm. *Syst. Sci. Control Eng.* **2020**, *8*, 22–34. [CrossRef]
25. Zhang, X.; Xiao, F.; Tong, X.; Yun, J.; Liu, Y.; Sun, Y.; Tao, B.; Kong, J.; Xu, M.; Chen, B. Time Optimal Trajectory Planing Based on Improved Sparrow Search Algorithm. *Front. Bioeng. Biotechnol.* **2022**, *10*, 852408. [CrossRef]
26. Liu, G.; Shu, C.; Liang, Z.; Peng, B.; Cheng, L. A Modified Sparrow Search Algorithm with Application in 3d Route Planning for UAV. *Sensors* **2021**, *21*, 1224. [CrossRef]
27. Zhou, S.; Xie, H.; Zhang, C.; Hua, Y.; Sui, X. Wavefront-Shaping Focusing Based on a Modified Sparrow Search Algorithm. *Optik-Int. J. Light. Electron. Opt.* **2021**, *244*, 167516. [CrossRef]
28. Khedr, A.M.; Vijayan, D.; Salim, A.; Elsway, A.A.; Osamy, W. ESSAIoV: Enhanced Sparrow Search Algorithm-Based Clustering for Internet of Vehicles. *Arab. J. Sci. Eng.* **2023**, 1–27. [CrossRef]

29. Zhang, Z.; He, R.; Yang, K. A Bioinspired Path Planning Approach for Mobile Robots Based on Improved Sparrow Search Algorithm. *Adv. Manuf.* **2022**, *10*, 114–130. [CrossRef]
30. Geng, J.; Sun, X.; Wang, H.; Bu, X.; Liu, D.; Li, F.; Zhao, Z. A Modified Adaptive Sparrow Search Algorithm Based on Chaotic Reverse Learning and Spiral Search for Global Optimization. *Neural Comput. Appl.* **2023**. [CrossRef]
31. Khaleel, M.I. Efficient Job Scheduling Paradigm Based on Hybrid Sparrow Search Algorithm and Differential Evolution Optimization for Heterogeneous Cloud Computing Platforms. *Internet Things* **2023**, *22*, 100697. [CrossRef]
32. Lai, X.; Wu, D.; Wu, D.; Li, J.H.; Yu, H. Enhanced DWA Algorithm for Local Path Planning of Mobile Robot. *Ind. Robot. Int. J. Robot. Res. Appl.* **2023**, *50*, 186–194. [CrossRef]
33. Han, S.; Wang, L.; Wang, Y.; He, H. A Dynamically Hybrid Path Planning for Unmanned Surface Vehicles Based on Non-Uniform Theta* and Improved Dynamic Windows Approach. *Ocean Eng.* **2022**, *257*, 111655. [CrossRef]
34. Liu, Q.; Zhang, Y.; Li, M.; Zhang, Z.; Cao, N.; Shang, J. Multi-UAV Path Planning Based on Fusion of Sparrow Search Algorithm and Improved Bioinspired Neural Network. *IEEE Access* **2021**, *9*, 124670–124681. [CrossRef]
35. Khedr, A.M.; Al Aghbari, Z.; Raj, P.P.V. MSSPP: Modified Sparrow Search Algorithm Based Mobile Sink Path Planning for WSNs. *Neural Comput. Appl.* **2023**, *35*, 1363–1378. [CrossRef]
36. Moysis, L.; Petavratzis, E.; Volos, C.; Nistazakis, H.; Stouboulos, I. A Chaotic Path Planning Generator Based on Logistic Map and modulo Tactics. *Robot. Auton. Syst.* **2020**, *124*, 103377. [CrossRef]
37. Li, Y.; Han, M.; Guo, Q. Modified Whale Optimization Algorithm Based on Tent Chaotic Mapping and Its Application in Structural Optimization. *KSCE J. Civ. Eng.* **2020**, *24*, 3703–3713. [CrossRef]
38. Gupta, M.; Gupta, K.K.; Khosravi, M.R.; Shukla, P.K.; Kautish, S.; Shankar, A. An Intelligent Session Key-Based Hybrid Lightweight Image Encryption Algorithm Using Logistic-Tent Map and Crossover Operator for Internet of Multimedia Things. *Wirel. Pers. Commun.* **2021**, *121*, 1857–1878. [CrossRef]
39. Ewees, A.A.; Abd Elaziz, M.; Houssein, E.H. Improved Grasshopper Optimization Algorithm Using Opposition-Based Learning. *Expert Syst. Appl.* **2018**, *112*, 156–172. [CrossRef]
40. Zhang, S.; Luo, Q.; Zhou, Y. Hybrid Grey Wolf Optimizer Using Elite Opposition-Based Learning Strategy and Simplex Method. *Int. J. Comput. Intel. Appl.* **2017**, *16*, 1750012. [CrossRef]
41. Mirjalili, S. SCA: A Sine Cosine Algorithm for Solving Optimization Problems. *Knowl.-Based Syst.* **2016**, *96*, 120–133. [CrossRef]
42. Liu, M.; Yao, X.; Li, Y. Hybrid Whale Optimization Algorithm Enhanced with Lévy Flight and Differential Evolution for Job Shop Scheduling Problems. *Appl. Soft Comput.* **2020**, *87*, 105954. [CrossRef]
43. Chang, L.; Shan, L.; Jiang, C.; Dai, Y. Reinforcement Based Mobile Robot Path Planning with Improved Dynamic Window Approach in Unknown Environment. *Auton. Robot.* **2021**, *45*, 51–76. [CrossRef]
44. Li, C.; Huang, X.; Ding, J.; Song, K.; Lu, S. Global Path Planning Based on a Bidirectional Alternating Search A* Algorithm for Mobile Robots. *Comput. Ind. Eng.* **2022**, *168*, 108123. [CrossRef]
45. Hou, W.; Xiong, Z.; Wang, C.; Chen, H. Enhanced Ant Colony Algorithm with Communication Mechanism for Mobile Robot Path Planning. *Robot. Auton. Syst.* **2022**, *148*, 103949. [CrossRef]
46. Xu, C.; Zhu, H.; Zhu, H.; Wang, J.; Zhao, Q. Improved RRT$*$ Algorithm for Automatic Charging Robot Obstacle Avoidance Path Planning in Complex Environments. *Comput. Model. Eng. Sci.* **2023**, *137*, 2567–2591. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.